



الإدارة العامة للبحوث

أساليب البرمجة الحديثة بلغز يلسك

تأليف:

د. محمد عبد الله
م. محمد عبد الله

م. محمد عبد الله
م. محمد عبد الله

٢٠١٩/٥/٢٨

١٤	قائمة الأشكال
٢٠	قائمة الجداول
٢١	مقدمة الكتاب
٢٧	الجزء الأول... مقدمة عامة: الحاسب والبرمجة .
٢٩	الفصل الأول - الحاسب الآلى: العناصر والوظائف . (The computer : components and Functions)
٢٩	— وظائف الحاسب الآلى .
٣٨	— الهيكل الهرمى للبيانات .
٣٩	— أساليب استرجاع البيانات (File Access Methods) .
٤٣	— نظرية النظام وتطبيقاتها مع الحاسب الآلى .
٤٥	— عناصر (مكونات) نظام الحاسب الآلى .
٥٠	— أقسام الحاسبات الآلية حسب حجمها .
٥٥	— تمارين .

الفصل الثاني: مقدمة عن البرمجة التركيبية الحديثة. ٥٧

(Structured Programing)

- ٥٧ — أساليب البرمجة التقليدية .
- ٥٩ — أساليب البرمجة التركيبية الحديثة .
- ٦٢ — خصائص وفوائد البرمجة التركيبية .
- ٦٣ — المشاكل التي تعترض تطبيق البرمجة التركيبية .
- ٦٥ — أمثلة مبسطة عن أساليب البرمجة التركيبية .
- ٦٧ — تمارين .

الفصل الثالث: مقدمة عن التجزئة الوظيفية

٦٩ الهيكلية (الهرمية)

(Top - Down Functional Decomposition)

- ٧٠ — أسس التجزئة الوظيفية الهرمية :
- ٧٠ — الأساس الأول : معرفة الوظائف .
- ٧٠ — طبيعة دور الوظائف والأجزاء الخاصة بها .
- ٧٢ — متى تتوقف التجزئة .
- ٧٣ — طبيعة التفاعل فيما بين الأجزاء .
- ٧٤ — حجم الجزء .
- ٧٥ — الأساس الثاني : التركيب الهرمي (Top-Down) .
- ٧٥ — طرق رسم الهيكل الهرمي .

الصفحة	
٧٩	— الأجزاء التي يتكرر تنفيذها .
٨٠	— البيانات المشتركة بين الأجزاء .
٨١	— التسلسل في تنفيذ الأجزاء .
٨٢	— خصائص التجزئة الوظيفية الهرمية .
٨٢	— فوائد التجزئة الوظيفية الهرمية .
٨٦	— أمثلة عملية .
٩١	— تمارين .

الفصل الرابع: مقدمة عن وسائل تصنيف البيانات والمعلومات في الحاسب

٩٣	
٩٣	— وصف عام لأسلوب تعامل الحاسب مع البيانات .
٩٧	— أصناف البيانات .
١٠٠	— أصناف العبارات التوضيحية .
١٠٣	— المعالجات الأساسية التي يقوم بها الحاسب .
١١٠	— طبيعة التخاطب مع الحاسب .
١١٥	— تمارين .

الجزء الثاني: أساسيات استخدام لغة بيك

١١٧	
١١٩	الفصل الخامس: برمجة معالجات بسيطة
١١٩	— حالات التخاطب مع الحاسب .

الصفحة	
١٢١	— معالجة عمليات حسابية بسيطة .
١٢٨	— أشكال أخرى للتعليمات المشروحة .
١٢٨	— تعليمة الإسناد (LET) .
	— تعليمات وأساليب إيضاح المخرجات على شاشة العرض
١٣٠	(PRINT, TAB)
١٣٥	— مثال متكامل لمعالجة عملية حسابية بسيطة .
١٣٧	— معالجة عمليات حسابية مع أقواس .
١٤٠	— تمارين .

الفصل السادس: برمجة معادلات بسيطة مع عمليات إدخال ١٤٣

١٤٣	— مقدمة عن عمليات الإدخال إلى الحاسب .
	— تطوير برنامج لحل مشكلة حسابية باستخدام تعليمتى
١٤٥	(READ/DATA) .
	— تطوير برنامج باستخدام بيانات حرفية (غير رقمية) وتعريف
١٥١	المخرجات .
١٥٤	— تطوير برنامج باستخدام تعليمة أدخل (INPUT) .
١٥٩	— تمارين .

الفصل السابع: الدوارة البسيطة والتحكم بها ١٦١ (Simple Loops)

١٦١	— مقدمة عن استعمال الدوارة في معالجة البيانات .
-----	---

الصفحة	
١٦٣	— تطوير برنامج بواسطة تعليمة الذهاب إلى رقم سطر معين.
١٦٥	— كيفية عمل تعليمة (إذا/ إذن IF/THEN) بشكلها المبسط.
١٦٧	— تطوير برنامج باستخدام (IF/THEN) لإيقاف التنفيذ طبيعياً.
١٧٩	— تمارين.

١٨٣ **الفصل الثامن: تعليمات التشعب** (Branching Statement)

١٨٣	— مقدمة عن ضرورة التشعب لمعالجة البيانات.
١٨٥	• التشعب الثنائي.
١٨٥	— تطوير برنامج باستخدام تعليمة (IF/THEN).
١٩١	— تطوير برنامج باستخدام تعليمة (IF/THEN/ELSE).
١٩٤	— ملخص ما تقدم عن استخدامات تعليمة (IF/THEN).
١٩٥	— تطوير برنامج باستخدام تعليمة (IF/THEN) والمقارنة (OR, AND).
٢٠١	— أساليب تمثيل المقارنات المركبة.
٢٠٣	• التشعب المتعدد.
	— تطوير برنامج بواسطة تعليمة التشعب المتعدد (ON/GOTO)، بناء
٢٠٣	على قيمة معطاة.
	— تطوير برنامج باستخدام تعليمة (ON/GOTO) مع إجراء عمليات
٢٠٨	حسابية للتوصل إلى قيم التفرع.
	— كيفية استخدام تعليمة (اذهب إلى برنامج فرعى، والعودة منه
٢١٢	.(ON/GOSUB, RETURN)

الصفحة	
٢١٣	— مقارنة تعليمتى (ON/GOSUB, ON/GOTO) .
	— ملاحظات على استخدامات تعليمتى
٢١٤	(ON/GOSUB و ON/GOTO) .
٢١٥	— تطوير برنامج باستخدام تعليمتى (RETURN, ON/GOSUB) .
٢٢٠	— تمارين .

الفصل التاسع: استخدام الدالات (Functions)

٢٢٥ فى العمليات الحسابية

٢٢٦	• الدالات المبرمجة المبنية (BUILT-in) .
٢٢٦	— مقدمة عن الدالات المبرمجة .
٢٢٧	— بعض الدالات المبرمجة والمزودة فى معظم الحاسبات .
٢٣١	— تطوير برامج باستخدام الدالات المبرمجة .
٢٤٦	• الدالات المعرفة من قبل المبرمج .
٢٤٦	— مقدمة عن الدالات المعرفة .
٢٤٧	— تطوير برنامج باستخدام الدالة المعرفة .
٢٥١	— تمارين .

الفصل العاشر: التحكم فى طبع البيانات المخرجة

٢٥٣ والمعلومات

٢٥٣	— استعمالات تعليمية (اطبع) باستخدام (PRINT USING) .
-----	---

الصفحة	
٢٥٥	• البيانات العددية .
٢٥٥	— طباعة الأرقام الصحيحة .
٢٥٥	— طباعة قيم عددية تحتوى على كسور عشرية .
٢٥٧	• الرموز الخاصة .
٢٥٨	• البيانات غير العددية .
٢٥٩	• العناوين التعريفية .
٢٦٠	— ملاحظات على استخدام تعليمة (PRINT USING) .
٢٦١	— تطوير برنامج باستخدام (PRINT USING) .
٢٦٤	• تصميم المخرجات والمدخلات .
٢٦٤	— المخرجات .
٢٧٠	— المدخلات .
٢٧١	— التحكم فى استقبال أو طبع المعلومات فى أى مكان على الشاشة .
٢٧٤	— لائحة الاختيارات الهرمية وتصميم الشاشات .
٢٧٤	— أنواع ووظائف الشاشات الرئيسية .
٢٧٥	— تطوير برنامج باستخدام لائحة الاختيارات الهرمية .
٢٨٨	— تمارين .

الفصل الحادى عشر: المصفوفات ذات البعد الواحد (One Dimensional Array)

٢٩٣ وتعليمات الدوارة البسيطة.

٢٩٣	— مقدمة عن المصفوفات ذات البعد الواحد .
-----	---

الصفحة	
٢٩٧	— تطوير برنامج باستخدام المصفوفات ذات البعد الواحد .
٣٠٣	— مقدمة عن الدوارة البسيطة .
	— تطوير برنامج باستخدام تعليمتى (FOR/NEXT) للتحكم فى الدوارة البسيطة .
٣٠٥	
	— تطوير برنامج باستخدام (FOR/NEXT) لمعالجة المصفوفات ذات البعد الواحد .
٣١٤	
٣٢٣	— تمارين .

الفصل الثانى عشر: المصفوفات ذات البعدين

(Two Dimensional Arrays)

ومكوناتها وتعليمات الدوارة

المتقدمة	٣٢٧
— مقدمه عن المصفوفات ذات البعدين .	٣٢٧
— العلاقة بين أرقام الأسطر والأعمدة .	٣٣١
— المجاميع الأفقية والعمودية .	٣٣٢
— مقدمة عن الدوارة المركبة (nested loops) وكيفية استخدامها فى عمليات المصفوفات ذات البعدين .	٣٣٣
— تطوير برنامج باستخدام تعليمات الدوارة المركبة مع المصفوفات ذات البعدين .	٣٤١
— إجراء العمليات الرياضية على المصفوفات واستخدامات تعليمة مصفوفة (MAT) .	٣٥٦

الصفحة	
٣٥٦	— تطوير برنامج باستخدام تعليمة (MAT) .
	— العمليات الرياضية التي يمكن إجراؤها على المصفوفات باستخدام
٣٦١	تعليمة (MAT) .
	— تطوير برنامج تجارى باستخدام تعليمات الدوارة المركبة
٣٧٠	وتعليمة (MAT) .
٣٧٤	— تمارين .

الجزء الثالث: مواضيع متقدمة فى لغة بيسك ٣٧٩

الفصل الثالث عشر: استخدام الملفات فى معالجة البيانات

٣٨١ File Processing

٣٨١	— مقدمة عن معالجة الملفات .
٣٨٢	— طرق التعامل مع الملفات .
٣٨٣	— فوائد استخدام الملفات .
٣٨٤	— تعليمات معالجة الملفات بالطريقة التتابعية .
٣٨٧	— تطوير برامج لاستخدام الملفات بالطريقة التتابعية .
٤١٣	— تعليمات معالجة الملفات بالطريقة العشوائية .
٤٢٠	— تطوير برنامج لمعالجة الملفات بالطريقة العشوائية .
٤٣٤	— تمارين .

الصفحة _____

الفصل الرابع عشر: تطبيقات (Applications) ٤٣٧

- ٤٣٧ — الفرز والدمج والبحث .
 - ٤٥٧ — تطبيق حكومي .
 - ٤٧٠ — تطبيق تعليمي .
 - ٤٨٥ — تمارين .
-

الفصل الخامس عشر: السلاسل: أساليب التعامل

٤٨٧ معها وتطبيقاتها

- ٤٨٧ — مقدمة عن السلاسل (STRINGS) .
 - ٤٨٧ — تعليمات السلاسل .
 - ٤٩٥ — تطوير برامج باستخدام تعليمات السلاسل .
 - ٥١٣ — تعليمات أخرى للتعامل مع السلاسل .
 - ٥١٩ — تمارين .
-

الفصل السادس عشر: الرسومات البيانية (Graphics) ٥٢٣

- ٥٢٣ — مقدمة عن الرسومات البيانية .
 - ٥٢٣ — فوائد استخدام الرسومات البيانية .
 - ٥٢٤ — أوضاع الشاشة .
 - ٥٢٦ — التحكم في أوضاع الشاشة .
-

الصفحة	
٥٢٧	— تطوير برامج على الرسومات البيانية .
٥٣٦	— مقدمة عن استخدام الألوان في الرسومات .
٥٣٧	— تطوير برنامج باستخدام تعلية (لون COLOR) .
٥٣٨	— تعليمات رسم الأشكال الهندسية مسبقة التحديد .
٥٤٥	— تطوير برنامج باستخدام تعلية (دائرة CIRCLE) .
٥٥٠	— تمارين .

ملحق عمليات التشغيل

٥٥٢	— مقدمة عن عمليات التشغيل .
٥٥٣	— خطوات التعامل مع الحاسبات الكبيرة .
٥٥٦	— خطوات التعامل مع الحاسبات الآلية الشخصية .
٥٦٢	— الإجراءات الخاصة بطباعة البرامج المطورة في الكتاب .
	— مقارنة لغة بيسك بلغتى البرمجة شائعتى الاستخدام
٥٦٤	(فورتران، كربول) .

قائمة الأشكال

الصفحة	العنوان	رقم الشكل
٤٤	أنواع أجهزة التخزين العشوائى والمتتالى .	١-١
٤٦	مكونات الحاسب الآلى وتفاعل بعضها مع بعض .	٢-١
٥٩	الأنماط الأربعة الأساسية للبرمجة التركيبية .	١-٢
٦٠	المرج المبسط بين الأنماط الأساسية للبرمجة التركيبية الحديثة .	٢-٢
٦١	المرج المعقد بين جميع الأنماط الأساسية للبرمجة التركيبية الحديثة .	٣-٢
٦٣	مراحل استخدام لغات البرمجة وأساليب البرمجة التركيبية .	٤-٢
٧٥	مثال تجريدى لهيكل هرمى .	١-٣
٧٧	المستوى الشمولى الأول فى التركيب الهرمى .	٢-٣
	التركيب التفصيلى الهرمى للجزء رقم (١٠١ هـ) من الشكل	٣-٣
٧٧	(٢-٣) .	
٧٨	تركيب هرمى ذو تفرعات أفقية متعددة .	٤-٣
٧٨	تفصيل الجزء (١ س) التابع لشكل (٣-٤) .	٥-٣
	استخدام مزيج من الأساليب للإشارة إلى الأجزاء المتكررة	٦-٣
٨٠	التي تقوم بنفس الوظيفة فى تركيب هرمى تفصيل .	
٨٣	الاختيار من أسفل إلى أعلى .	١٦-٣
٨٦	الهيكل الهرمى العام لوظائف الإنسان .	٧-٣
٨٧	الهيكل الهرمى التفصيل للجزء (١٠١) - «تناول الطعام» .	٨-٣
٨٨	الهيكل الهرمى لوظائف المكتبة .	٩-٣

تابع قائمة الأشكال

الصفحة	العنوان	رقم الشكل
٨٨	الهيكل الهرمي التفصيلي للجزء (٢٠٤) — «تحديث الملف الرئيسي» .	١٠ — ٣
٨٨	الهيكل الهرمي التفصيلي للجزء (٣) — «تجديد الكتاب» .	١١ — ٣
٨٩	الهيكل الهرمي العام لمعالجة الحسابات الجارية الخاصة بالبنك (س ص ع) .	١٢ — ٣
٩٠	الهيكل الهرمي التفصيلي للجزء (١٠٢) — «الحسابات الجديدة» .	١٣ — ٣
٩٠	الهيكل الهرمي التفصيلي للجزء (٢٠٤) — «تحضير القوائم» .	١٤ — ٣
١١١	أسلوب التخاطب مع الحاسب عند تطوير برنامج معين .	١ — ٤
١٢٧	استخدام الهيكل الهرمي لكتابة برنامج لحساب مساحة السجاد المبيع .	١ — ٥
١٣٦	استخدام الهيكل الهرمي لكتابة برنامج لحساب مساحة المعين .	٢ — ٥
١٣٩	برنامج لحساب محيط ومساحة مثلث إذا علم أطوال أضلاعه .	٣ — ٥
١٤٦	برنامج لحساب العمولة لمندوبي المبيعات .	١ — ٦
١٤٩	برنامج لحساب معدل خمس علامات .	٢ — ٦
١٥٣	برنامج لحساب معدل خمس علامات لدارس واحد مع طباعة الاسم والعناوين .	١ — ٦
١٥٥	برنامج لحساب مساحة ومحيط مستطيل إذا علم طوله وعرضه .	٣ — ٦
١٦٤	برنامج لحساب محيط الدائرة باستخدام عبارة (GOTO) .	١ — ٧

تابع قائمة الأشكال

الصفحة	العنوان	رقم الشكل
	برنامج لحساب محيط الدائرة باستخدام تعليمة (إذا/ إذن) لإيقاف	٧ - ٢
١٧١	البرنامج بواسطة القيمة الوهمية ورسم البرمجة التركيبية .	
	برنامج لحساب محيط الدائرة وإيقاف التنفيذ عن طريق العداد	٧ - ٣
١٧٥	ورسم البرمجة التركيبية .	
	أسلوب آخر لوقف تنفيذ البرنامج عند نفاذ القيم المدخلة	٧ - ٣ أ
١٧٧	ورسم البرمجة التركيبية .	
	برنامج لحساب الراتب الإجمالي لموظفى مؤسسة معينة باستخدام	٨ - ١
١٨٧	عبارة (IF/THEN) واحدة .	
	برنامج لحساب الراتب الإجمالي لموظفى مؤسسة معينة باستخدام	٨ - ١ أ
١٨٩	عبارات (IF / THEN) متعددة .	
	برنامج لحساب الرواتب الإجمالية لموظفى مؤسسة معينة	٨ - ١ ب
١٩٣	باستخدام تعليمة (IF/THEN/ELSE) .	
	برنامج لإيجاد عدد الشيكات المحولة للموظف (x) والموظف (y)	٨ - ٢
٢٠٠	والمعاداة .	
	برنامج لإيجاد المبلغ الإجمالى فى نهاية الأسبوع لمبيعات	٨ - ٣
٢٠٧	صنفين من البضاعة .	
٢١٠	برنامج لإيجاد عدد الدارسين لمواد الحاسب الآلى .	٨ - ٤
	برنامج لحساب الدخل الإجمالى لمؤسسة تأجير سيارات ،	٨ - ٥
٢١٨	والدخل التفصيلى لفروعها .	

تابع قائمة الأشكال

الصفحة	العنوان	رقم الشكل
٢٣٣	برنامج لإيجاد عوامل عدد باستخدام الدالة (INT) .	١ - ٩
	برنامج للحصول على عشرة أرقام عشوائية باستخدام	٢ - ٩
٢٣٦	الدالة (RND) .	
	برنامج للحصول على ثلاث مجموعات من الأرقام	٣ - ٩
٢٤٠	العشوائية الصحيحة .	
٢٤٤	برنامج خاص بعمليات انتظار الزبائن أمام محطات البنزين .	٤ - ٩
	برنامج لتحويل الدرجات المئوية إلى الفهرنهايتية وبالعكس	٥ - ٩
٢٤٩	باستخدام الدالة المعرفة DEF .	
	برنامج لحساب الراتب الأساسي وخارج الدوام باستخدام	١ - ١٠
٢٦٢	(PRINT USING)	
	برنامج لإيجاد الدخل الكلي لكل فرع من فروع الشركة	٢ - ١٠
٢٨٠	والدخل الإجمالي للشركة .	
	برنامج لحساب صافي أرباح بائع الجرائد أسبوعياً ونسبة	١ - ١١
٣٠٢	ربح كل يوم إلى مجموع الأرباح .	
	برنامج لحساب قيمة الاستهلاك المتناقصة وفق أسلوب	٢ - ١١
٣١٠	عدد أرقام السنوات .	
	برنامج لتصنيف الأعمار إلى فئات أربع ، وطباعة أعمار	٣ - ١١
٣٢١	كل فئة مع عددها ونسبتها .	

تابع قائمة الأشكال

الصفحة	العنوان	رقم الشكل
٣٤٨	برنامج لحساب عدد الساعات المستأجرة من قبل الأندية لصالات المدينة الرياضية .	١٢ - ١
٣٥٢	برنامج لقراءة البيانات في مصفوفة ذات بعدين ، وحساب مجاميع الأسطر والأعمدة ، وطباعة المصفوفة مع النتائج .	١٢ - ١ أ
٣٥٨	برنامج لإيجاد متوسط درجات الحرارة لكل أسبوع ومن ثم الأسابيع الأربعة .	١٢ - ٢
٣٧٢	برنامج لإيجاد سعر البيع للوحدة من الأصناف الثلاثة المنتجة .	١٢ - ٣
٣٨٩	برنامج لإنشاء ملف مع بيانات عن مستودع رياضي .	١٣ - ١
٣٩٥	برنامج للاستفسار عن ثمن بضائع مستودع رياضي باستخدام الملفات .	١٣ - ٢
٣٩٨	برنامج لإجراء التعديلات على سجلات الملف المتتابع للبضائع .	١٣ - ٣
٤٠٤	برنامج لقراءة سجلات ملف ، وطباعتها .	١٣ - ٤
٤٠٩	برنامج لدمج ملف بيانات جديدة مع ملف البيانات الرئيسية وفهرزها ، ومن ثم كتابتها على الملف الرئيسي .	١٣ - ٥
٤٢٤	برنامج لإضافة وتعديل وحذف وطباعة بيانات مستودع رياضي باستخدام الطريقة العشوائية .	١٣ - ٦
٤٤٠	برنامج للحصول على أرقام عشوائية باستخدام دالة (RND) ، وفهرزها/ باستخدام أسلوب (BUBBLE) .	١٤ - ١

تابع قائمة الأشكال

الصفحة	العنوان	رقم الشكل
٤٤٥	برنامج للحصول على أرقام عشوائية باستخدام دالة (RND)، وفرزها باستخدام أسلوب (SHELL) .	٢ — ١٤
٤٥٠	برنامج للحصول على أرقام عشوائية باستخدام دالة (RND)، وفرزها باستخدام أسلوب (HEAP) هيب .	٣ — ١٤
٤٥٦	برنامج للبحث عن قيمة في مصفوفة مفروزة باستخدام أسلوب البحث الثنائي (BINARY SEARCH) .	٤ — ١٤
٤٦١	برنامج لدمج مجموعتين من الدارسين ، وإيجاد المعدل العام والتقدير لكل دارس ومن ثم إمكانية الاستفسار .	٥ — ١٤
٤٧٥	برنامج لإظهار الأجزاء الرئيسية للحاسب وانسياب البيانات وكيفية التحكم .	٦ — ١٤
٤٩٨	برنامج للاستفسار عن عدد القطع الموجودة من البديل الرجالية .	١ — ١٥
٥٠٤	برنامج لتحويل مجموعة من الأعداد بالعربية إلى ما يرادفها بالهندية	٢ — ١٥
٥١٠	برنامج لتحويل نص معين إلى نص الشفرة باستخدام طريقة الأعمدة .	٣ — ١٥
٥٢٩	برنامج لإظهار المبيعات الشهرية على شكل أعمدة .	١ — ١٦
٥٣٣	برنامج لإظهار المسافة اللازمة لإيقاف السيارة حسب سرعتها .	٢ — ١٦
٥٤٧	برنامج لإظهار مبيعات الشركة للثاني عشر شهراً برسم الدائرة المقسمة .	٣ — ١٦

قائمة الجداول

الصفحة	العنوان	رقم الجدول
٥٦٠	بعض العمليات والتعليمات المتعلقة بكل منها في الحاسبات الآلية الشخصية .	١٧ - أ
٥٦١	بعض العمليات والتعليمات المتعلقة بكل منها في الحاسبات الآلية الكبيرة .	١٧ - ب
٥٦٦	مقارنة بين التعليمات والقواعد الخاصة بها في لغات البرمجة (بيسك، فورتران، كوبول).	١٧ - ج

مقدمة الكتاب

زاد الحديث في السنوات الأخيرة عن أهمية الأساليب الحديثة في البرمجة، وبشكل خاص البرمجة التركيبية والهيكل الهرمي؛ لما لذلك من أثر كبير في زيادة إنتاجية المبرمج وتحسين وضوح وصيانة البرامج. كذلك فإن لغة بيسك أصبحت شائعة الاستعمال، بسبب كونها اللغة الأساسية التي تتعامل معها الحاسبات الشخصية واسعة الانتشار. لذلك يهدف هذا الكتاب إلى تقديم لغة بيسك وطرق استخدامها، وفق الأساليب الحديثة للبرمجة، ولتحقيق هذا الهدف، تم استخدام برنامج BASICA على جهاز IBM/XT، حيث تتم معالجة البيانات المعطاة باللغة العربية بواسطة تعليمات لغة بيسك باللغة الإنجليزية.

يقع الكتاب في ستة عشر فصلاً، جُزئت إلى أجزاء رئيسية ثلاثة، بالإضافة إلى ملحق عن عمليات تشغيل لغة بيسك على بعض الحاسبات الشخصية الكبيرة، بحيث يطلع المستخدم على الأجزاء التي تهتمه، وتناسب مستواه فقط. فلم يتم مزج شروحات الأساليب الحديثة وتعليمات البرمجة، بينما تم التقديم لكيفية عمل بعض التعليمات قبل شرحها واستخدامها.

ويمكن تقسيم الكتاب إلى الأجزاء التفصيلية التالية:

- (١) مقدمة عن الحاسب الآلي : وظائفه وأجزائه.
- (٢) مقدمة عن الأساليب الحديثة.
- (٣) مقدمة عن كيفية معالجة البيانات من قبل الحاسب في فصل مستقل، ومقدمات أخرى متناثرة قبل كل فصل أو موضوع له علاقة بالبرمجة، يستطيع ذو الخبرة في

البرمجة أن يتجاهلها ويستمر في القراءة دون أن يعطل ذلك الفهم والاستيعاب لتعليمات اللغة.

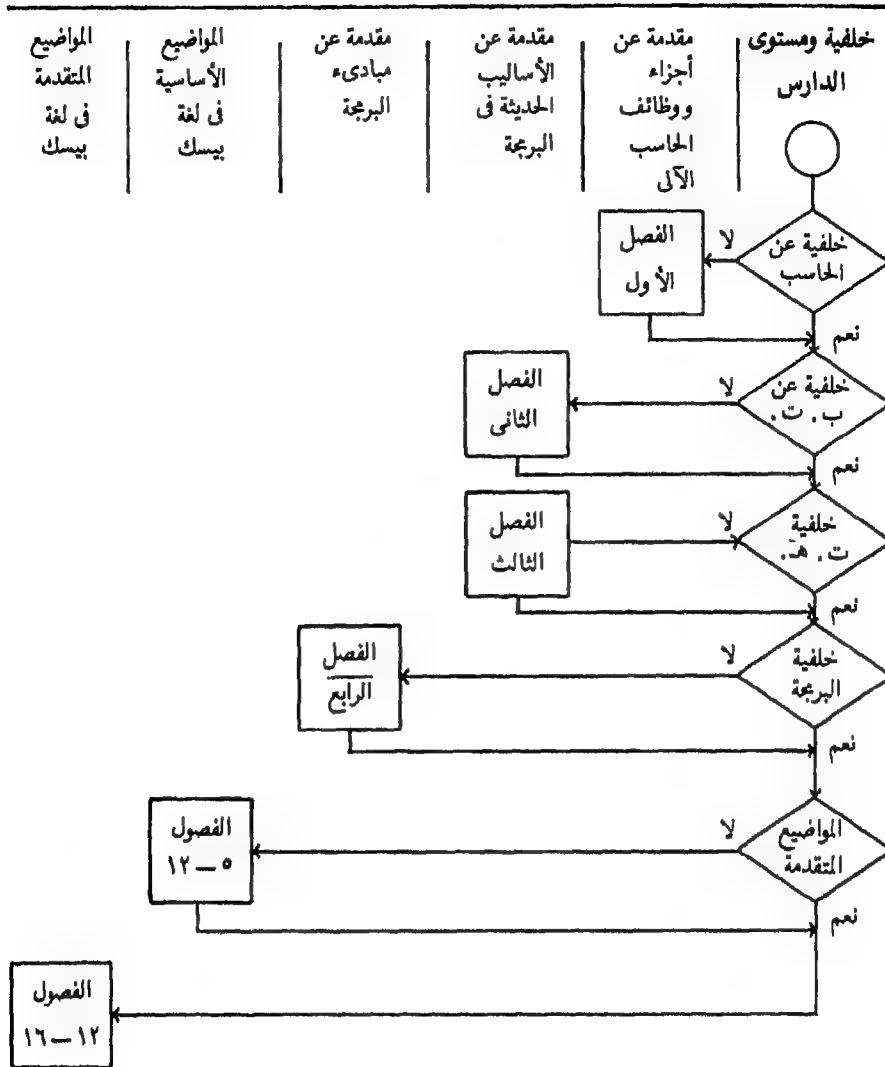
- ٤) مواضيع أساسية في كيفية استخدام تعليمات لغة بيسك.
- ٥) مواضيع متقدمة في كيفية استخدام تعليمات لغة بيسك.
- ٦) ملحق تعليمات التشغيل لبعض الحاسبات الكبيرة والصغيرة.

ويبين الرسم التالى كيفية اختيار مواضيع الكتاب للقراءة حسب مستوى وخلفية الدارس، فيما عدا الملحق، حيث يرجع إليه للحصول على تعليمات تشغيل لغة BASICA على جهاز IBM/PC، ولمعرفة الفروقات بين عدة أجهزة أخرى، إلى جانب IBM، في كيفية استخدامها للغة بيسك (BASIC)، ولمقارنة لغة بيسك بـ كوبرول وفورتران.

ويختلف أسلوب تقديم تعليمات لغة بيسك في هذا الكتاب عن غيره من الكتب الأخرى*. والأسلوب التقليدى هو التحدث عن قواعد التعليمات وأشكالها المختلفة. والصعوبة التى تواجه الدارس هى في كيفية ربط خطوات حل مشكلة بتسلسل التعليمات الخاصة بحلها، وكثيراً ما نسمع من الدارسين من يقول: «أنا أفهم كيف تعمل تعليمة اطبع PRINT أو أدخل INPUT إلا أنني أشعر بالجهل التام عندما أحاول ربط هذه التعليمات بأسلوب وخطوات حل مشكلة معينة».

أما هذا الكتاب فيبدأ من أمثلة بسيطة ومن ثم يقدم التعليمات الضرورية لحل المثال، وأحياناً قد يقدم لضرورة وجود تعليمة ذات وظيفة معينة من واقع أمثلة عملية. وبعد ذلك يتم شرح الأشكال الأخرى للتعليمة ومع أمثلة أخرى أكثر تعقيداً، إذا اقتضى الأمر.

* لقد جاء هذا الكتاب كثمرة لما يزيد على ١٥ سنة من الخبرة التدريسية والتعليمية والعملية للمؤلفين، في مجال تدريس لغة بيسك وكذا في مجال التحليل والتصميم وقواعد البيانات أيضاً.



ب.ت.ب. : برمجة تركيبية
 ت.ه.ب. : تجزئة هرمية وظيفية (هيكل هرمي)

لذا يمتاز أسلوب هذا الكتاب عن غيره من الكتب الأخرى بالخصائص التالية :

١ - وجود فصول متقدمة ومستقلة لشرح أسس استخدام الحاسب والأساليب الحديثة في البرمجة، بحيث يمكن استخدام هذه الفصول بشكل مستقل عند التدريب على أية لغة برمجة .

٢ - الانطلاق من التمارين والأمثلة وخطوات الحل إلى تعليمات بيسك في توضيح كيفية استخدام الأساليب الحديثة في البرمجة . وذلك حتى يتمكن الدارس من استخدام هذه التعليمات في تطبيقات مختلفة .

٣ - شرح تعليمة الذهاب إلى برنامج فرعى GOSUB في الفصول الأولى ، وذلك حتى يعتاد الدارس على كيفية استخدامها في برمجة أجزاء الهيكل الهرمي .

٤ - إجراء التعديلات والتطويرات (الصيانة) على البرامج المستخدمة لحل الأمثلة ، وذلك بتغيير أو تعقيد إجراءات وخطوات الحل فيها ؛ وذلك لإعطاء الدارس فكرة عن كيفية وأهمية إجراء هذه التعديلات على البرامج القائمة .

وقد اتبع الكتاب الخطوات الرئيسية التالية في معظم الحالات ، إلا إذا اقتضت الضرورة المزج بين هذه الخطوات ، إما لتشابه الأمثلة أو لزيادة توضيحها :

أولاً - بعد تحديد الهدف ، يتم سرد الخطوات الرئيسية اللازمة لتحقيق الهدف ، وأحياناً قد يربط بها تعليمات البرمجة المناسبة .

ثانياً - الهيكل الهرمي للبرنامج (TOP-DOWN) .

ثالثاً - نمط البرمجة التركيبية المناسب .

رابعاً - البرنامج الكامل وفق أساليب البرمجة الحديثة .

ولا يتطلب استخدام هذا الكتاب بالشكل الأمثل أية خلفيات أو معلومات مسبقة .

ونود أخيراً أن نوجه الشكر إلى زوجتينا وأولادنا الذين ظلوا حبيسي المسكن نهائياً
وليلاً في ظروف كانوا هم فيها أحوج ما يكونون إلى أن يمضوا بعض ذلك الوقت
خارجه ، وذلك كله حتى نستطيع إنجاز هذا الكتاب في المدة المحددة .

مقدمة عامة
الحاسب والبرمجة

الحاسب الآلي
(المناصر والوظائف).
مقدمة عن البرمجة
التركيبية الحديثة.
مقدمة عن التجزئة
الوظيفية الهيكلية (الهرمية)
مقدمة عن وسائل تصنيف
البيانات والمعلومات
في الحاسب.

الفصل الأول

الحاسب الآلى

العناصر والوظائف

وظائف الحاسب الآلى :

أراد الإنسان منذ القدم أن يستعمل أجهزة تساعده فى القيام ببعض الأعمال التى يقوم بها . وقد أولى الإنسان اهتماماً محدوداً فى العصور القديمة لعمليات العد الحسابى ، فكان استخدامه للأصابع والأحجار مثلاً ، ولكن الاهتمام الأكبر انصب على الأعمال التى لها علاقة بالأمور المعيشية والحياتية ، كتوفير المأكل والمشرب والمسكن . ونتج عن ذلك الشورتان الزراعـية والصناعية اللتان أدتا إلى استعمال الآلات لتحسين وزيادة الإنتاج الزراعى والصناعى .

ومع التقدم العلمى فى الزراعة والصناعة ، زاد التقدم العمرانى واتصال الأمم بعضها مع بعض ، وتعددت النشاطات والعلاقات بين الأفراد والمؤسسات والأمم ، وزادت المهام والوظائف التى تقوم بها الدول والخدمات التى تقدمها لرعائها ، فكان لابد من الحفاظ على معلومات عن هذه النشاطات والوظائف والخدمات ، بهدف تقويم نتائجها والتخطيط لمستقبلها .

وأول استخدام رئيسى للحاسب الآلى كان فى تعداد السكان للولايات المتحدة الأمريكية فى أواخر القرن التاسع عشر . وبالمقارنة بهذا المثال الحكومى ، فإن أول استخدام تجارى كان فى مجال المحاسبة ، حيث تطلب الأمر الحفاظ على معلومات عن العمليات الحاسبية بين صرف وإيراد واستهلاك ، وذلك حتى تعكس الصورة السليمة للوضع المالى للمؤسسات .

و يلاحظ من هذين الاستعمالين الأمور التالية :

- ١ - كثرة المعلومات المراد التعامل معها .
- ٢ - الحاجة إلى القيام بالعمليات الحسابية المطولة .
- ٣ - ضرورة القيام بالمقارنات بهدف التصنيف والفهرسة والتجميع والفرز .
- ٤ - الدقة والسرعة في إنجاز ماتقدم .

وبناء على ذلك فإن الحاسب يقوم بوظيفتين رئيسيتين بسرعة ودقة متناهيتين :

١ - العمليات الحسابية **Arithmetic operations** : يقوم الحاسب بجميع العمليات الحسابية من جمع وطرح وضرب وقسمة وأس ، مع العلم بأن الحاسب يقوم بجميع العمليات الحسابية بأسلوب الجمع .

٢ - العمليات المنطقية **Logical operations** : يقوم الحاسب بالعمليات المنطقية عن طريق إجراء المقارنات الثنائية بين معلومتين محددين ، بحيث لا تخرج نتيجة المقارنة في نطاق محدد عن أحد الاختيارين . فمثلا عند مقارنة تساوى معلومتين ، فنتيجة المقارنة تكون إما تساوى كليهما أو عدمه ، ولا يندرج صغر أو كبر إحداهما عن الأخرى تحت نطاق هذه المقارنة . وكذلك الأمر مع مقارنات «أصغر من» ، «أكبر من» «لا يساوى» و «أصغر من ويساوى» و «أكبر من ويساوى» . وتستخدم المقارنات هذه في كثير من الأحيان للتشعب من خطوة إلى أخرى .

ولا بد من التنويه هنا بأن الحاسب يقوم بوحدة من العمليات الحسابية والمنطقية فقط في أية لحظة من اللحظات الزمنية التي تمر بها العمليات المتعددة . إذ قد يتطلب استخراج نتائج مطلب معين ، القيام بعمليات حسابية مطولة ومقارنات متعددة ، فالحصول على أسماء الأفراد الذين يتجاوزون الستين من العمر وعددهم ، يتطلب التأكد من العمر - ن طريق المقارنة المنطقية ، ومن ثم حساب العدد الكلى بعد كل مقارنة إلى

أن تنتهى قائمة أفراد السكان . وقد يصل التعقيد والتداخل بين العمليات الحسابية والمنطقية درجة يصعب معها ملاحظة تسلسل العمليات يدوياً .
واستناداً إلى الوظيفتين الرئيسيتين : الحسابية والمنطقية ، قد يقوم الحاسب بالوظائف الفرعية التالية :

١ - التجميع والاختصار : إذ يمكن مثلاً الحصول على مجموع المبالغ المصروفة للرواتب ، ومجموع المبيعات لفترة سنة لمؤسسة ما ، أو مجموع عدد الموظفين في الدرجات المختلفة . وقد يكون التجميع والاختصار قد بنيا على مقارنات منطقية ، فمن الممكن مثلاً الحصول على عدد الأفراد الذين يقعون ضمن فئات محددة من الأعمار .

٢ - التبويب والتصنيف : إذ يمكن الحصول على المعلومات حسب أنواع البضائع المنتجة ، أو حسب المواد والقطع المخزنة ، أو حسب أنواع الحسابات ، أو حسب فئات المستهلكين ، أو حسب المواد الداخلة في التصنيع لسلعة ما .

٣ - الفرز والدمج : إذ يمكن الحصول على قائمة سلسلة حسب الحروف الأبجدية ، أو حسب القيم الرقمية تصاعدياً أو تنازلياً . وكذلك من الممكن دمج قائمتين مفروزتين إذا تشابهت طبيعة المعلومات حرفياً أو رقمياً ، وذلك بهدف إنتاج قائمة جديدة موحدة .

وقد تكون القائمة المفروزة بها أسماء موظفين ، أو الدرجات ومسمياتها ، أو المبيعات حسب المناطق ، أو أرقام القطع وكمياتها المخزنة ، أو درجات الدارسين في امتحان ما .

٤ - الفهرسة والتقسيم : إذ يمكن الحصول على المعلومات الخاصة بالسكان مقسمة حسب المناطق الجغرافية ، أو مجمل دخل مندوبى المبيعات حسب نوع السلعة التى يتعاملون معها ، أو تكلفة بناء المنازل حسب تصاميمها المختلفة .

ويلاحظ الفرق بين الفهرسة والتقسيم من جهة ، والتبويب والتصنيف من جهة أخرى : إن الأولى لا تتطلب وجود علاقة منطقية تربط بين الأقسام المختلفة ، بينما تتطلب الثانية علاقة يحكمها نظام معين للربط بين الأقسام المختلفة .

فالتصنيف قد يكون بناءً على علاقة هرمية ، فتصنيف المكتبات يقسم موضوعات الكتب إلى عشرة أقسام رئيسية ، ومن ثم يتم تصنيف الموضوعات الفرعية ضمن كل قسم رئيسي . فإذا أخذنا العلوم الاجتماعية كأحد الأقسام الرئيسية ، فتندرج تحته ، مثلاً : العلوم الاقتصادية ، وعلم النفس ، وإدارة الأعمال ، والإدارة العامة . ويشير التصنيف إلى مستويات تفصيلية حتى آخر مستوى .

أما تقسيم المساحات إلى مناطق جغرافية لا رابط بينها ، فهو فهرسة وليس تصنيفاً ، أو قد ترتب المناطق بطريقة عشوائية ، أو تفرز حسب أرقامها أو أسمائها .

وحتى يؤدي الحاسب الهدف من استخدامه ، لابد من قيامه بوظائف أخرى تكمل الوظائف المذكورة سابقاً .

وهذه الوظائف هي :

١ — عمليات الإدخال : إذ يجب أن يتم إدخال المعلومات والتعليمات الخاصة بهذه المعلومات ، والتي تحدد الخطوات التي ستخضع لها من عمليات حسابية ومنطقية ، وغيرها .

٢ — عمليات التخزين : وذلك لتخزين التعليمات والمعلومات المتعلقة بوصف نشاط معين . ويتم التخزين داخل الحاسب في نوعين من الذاكرة :

أ — الذاكرة الرئيسية (Main Memory) : وهي الذاكرة النشطة والمستعدة باستمرار لاستقبال المعلومات والتعليمات طالما كان الحاسب موصولاً بمصدر لتيار

كهربائى . وحال تخزين هذه المعلومات والتعليمات ، تكون قابلة لاتخاذ قرار
فورى بتشغيلها من قبل الحاسب .

ب - الذاكرة الفرعية (المساعدة) (Secondary Memory) : نجمت الحاجة إلى هذا
النوع من الذاكرة عن تضخم حجم الذاكرة الرئيسية مع ازدياد استخدام
الحاسب فى مجالات كثيرة، مما أدى إلى ازدياد التكلفة وصعوبة التحكم فى
التعليمات والمعلومات المختلفة، وبالإضافة إلى ذلك فإن بعض التعليمات
والمعلومات لانتحتاج إليها إلا على فترات متباعدة، لذلك كان لابد من إيجاد
ذاكرة مساعدة يتم فيها تخزين التعليمات والمعلومات إلى حين الحاجة إليها،
فحينئذ تنقل إلى الذاكرة الرئيسية .

٣ - عمليات الإخراج : وهى العمليات التى لها علاقة بإخراج نتائج تنفيذ
العمليات بطريقة تؤدى إلى الاستفادة من وظائف الحاسب من قبل المستخدمين .
ولنضرب مثلاً بسيطاً على هذه الوظائف . لنفترض أن شخصاً ما يريد أن يحسب
مساحة السجاد المعد للبيع . لتحقيق ذلك لابد من اتباع الخطوات التالية :

١ - الحصول على طول وعرض السجاد .

٢ - ضرب الطول فى العرض وتخزين الناتج .

٣ - كتابة الناتج .

فالخطوة الأولى تتطلب إدخالاً وتخزيناً ، والثانية تتطلب عمليات حسابية وتخزيناً ،
والثالثة تتطلب إخراجاً للنتائج . ومن ثم لابد من إدخال المعلومات المكونة من الطول
والعرض الفعليين :
٤٥ ، ٦٦ متراً مثلاً .

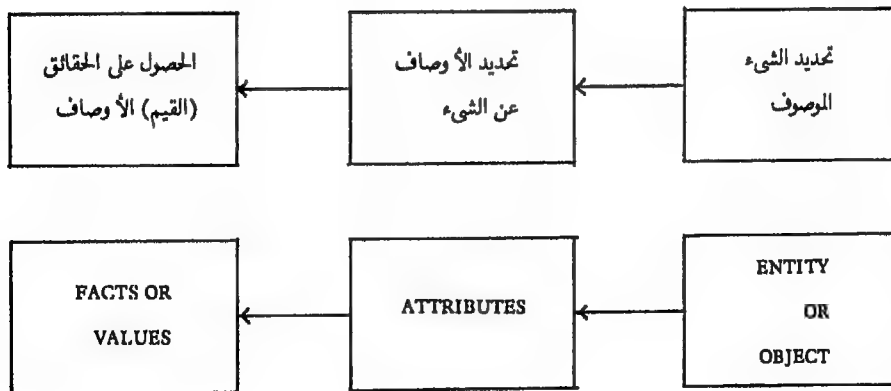
و يلخص الجدول التالى وظائف الحاسب الآلى :

الوظائف الفرعية	الوظائف الأساسية
١ - التجميع والاختصار .	١ - العمليات الحسابية .
٢ - التوزيع والتصنيف .	٢ - العمليات المنطقية .
٣ - الفرز والدمج .	الوظائف المكتملة :
٤ - الفهرسة والتقسيم .	١ - عمليات الإدخال
	٢ - عمليات التخزين .
	٣ - عمليات الإخراج .

البيانات والمعلومات والتعليمات :

تعكس هذه الكلمات الثلاث الوظائف التى يقوم بها الحاسب الآلى ، وسنبين معنى كل منها وعلاقتها مع هذه الوظائف وعلاقة بعضها ببعض .

البيانات : من الممكن تعريف البيانات بأنها : «حقائق (قيم) تعكس أوصافاً (خصائص) معينة عن شئ محدد» ؛ لذلك يكون التسلسل المنطقى للوصول إلى البيانات هو :

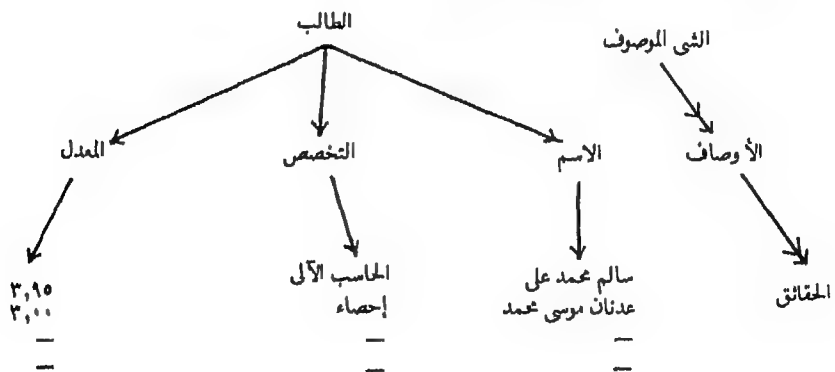


والشيء الموصوف Entity or object يمكن أن يكون مادياً ملموساً أو معنوياً ،
والملموس إما أن يكون من الأحياء أو من الجماد ، ومن الأمثلة على كل نوع :
الشيء الملموس في الأحياء : الموظف ، الطالب ، المدرس ، الشرطي ، الحيوان
الأليف ، الدابة .

الشيء الملموس من الجماد : أسهم الشركة ، البضاعة ، قطعة الغيار ، غرفة الدراسة ،
السيارة .
الشيء المعنوي : الشركة ، الطقس .

وأوصاف الأشياء تحدد بوظائفها ، فهي إما أن تكون أوصافاً لها علاقة بالعمليات
الحسابية ، كطول وعرض ووزن ودرجة حرارة ، وعمر وتاريخ البلاد ، أو لها علاقة
بالعمليات غير الحسابية ، كاسم الموظف وعنوانه ، ونوع الشركة ، ووصف قطعة الغيار .
والحقائق تتكون من أنواع الأحرف الثلاثة : الرقمية (٠ - ٩) ، الأبجدية (أ - ي) ،
والخاصة (مثل / ؟ ! ، ،) . فالأوصاف الحسابية يجب أن تتكون من أرقام
فقط ، وغير الحسابية يمكن أن تتكون من أى من أنواع الأحرف الثلاثة أو خليط من
اثنين أو أكثر . فمثلاً يكون العمر « ٢٥ » سنة ، والاسم « سلمان علي » والعنوان
« ص . ب . ١١١ ، الرياض ١١١٤٥ » ورقم القطعة « ص ت ج / ٢١٦٥ »

ويمكن توضيح العلاقة بين العناصر الثلاثة بالشكل الهرمي التالي :



و يلاحظ أحادية الشيء الموصوف ، وتعدد الأوصاف للشيء المحدد ، وتعدد الحقائق لكل وصف .

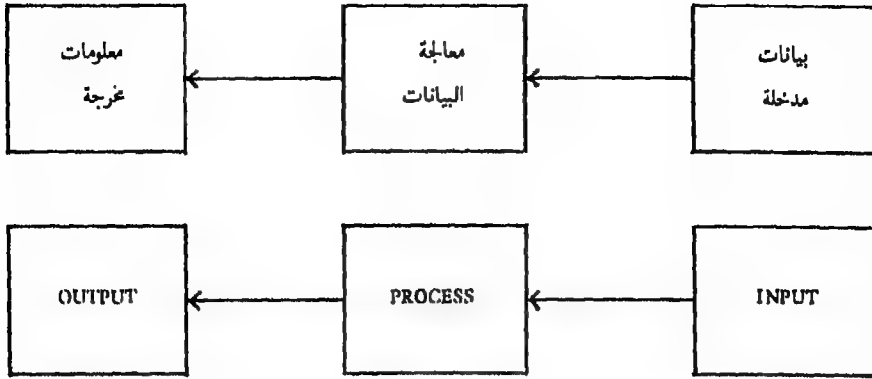
المعلومات : ويمكن تعريف المعلومات بأنها : البيانات التي تمت معالجتها حسابياً و/ أو منطقياً ، ومن ثم عرضها على المستفيد بأسلوب يحقق الاستفادة المرجوة منها . فقد سبق طباعة الشيك مثلاً عمليات حسابية للوصول إلى صافي الراتب من مجمله ، وعمليات منطقية لفرز الشيكات حسب رقم الموظف ، ومن ثم يتم اختيار الورق المناسب كوسيلة لصرف المبلغ من المصرف ، وتطبع عليه العبارات المناسبة ، مثل اسم المؤسسة وغيرها من العبارات . فمثلاً تطبع عبارات «معهد الإدارة العامة» و «ادفعوا لأمر» و «المدير المالي» و «أمين الصندوق» ، وأخيراً تتم طباعة الشيكات في نهاية كل شهر لتوافق الدورة المحاسبية ، وتعطى للموظف ليقوم بتسلم أو إيداع المبلغ .

و يلاحظ أن المعلومات تتألف من :

$$\left[\begin{array}{c} \text{العبارات} \\ \text{معهد الإدارة العامة} \\ \text{ص. ب ٢٠٥} \\ \text{ادفعوا الأمر} \end{array} \right] + \left[\begin{array}{c} \text{بيانات مخرجة} \\ \text{صافي الراتب} \\ \text{مجموع العلاوات} \\ \text{مجموع الخصومات} \end{array} \right] + \left[\begin{array}{c} \text{بيانات مدخلة} \\ \text{مجموع الراتب} \\ \text{العلاوات} \\ \text{الخصومات} \end{array} \right] = \text{المعلومات}$$

وبالإضافة إلى الطباعة على الورق ، يمكن استخدام الشاشة التلفزيونية (CRT) كوسيلة لعرض المعلومات .

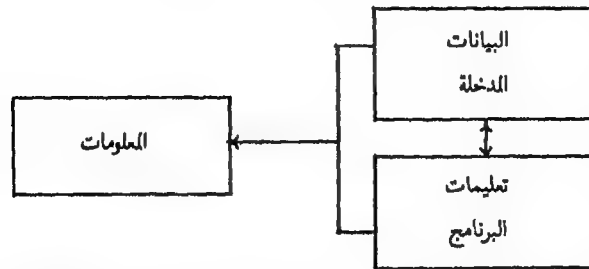
التعليمات : يمكن تعريف التعليمات بأنها : الخطوات التي يجب اتباعها لتحويل البيانات المدخلة إلى معلومات وفق الهدف المحدد ، وهذه الخطوات هي نفسها التي يقوم بها الإنسان للتوصل إلى المعلومات المطلوبة ، ولكنها قد تكون أكثر تحديداً وتفصيلاً عند توصيفها للحاسب . وتتبع هذه الخطوات الإطار العام التالي :



ولتحقيق هدف معين قد تتعدد وتتداخل البيانات المدخلة والمعالجات والمعلومات المخرجة . فالبيانات المدخلة قد تكون حقائق تعكس أشياء متعددة ، والمعالجات قد تكون حسابية و/ أو منطقية ، والمخرجات من معالجة معينة قد تؤخذ كلياً أو جزئياً كمدخلات لمعالجة أخرى ... وهكذا .

ولما كان التخاطب مع الحاسب بوضعه الحال لا يمكن أن يتم بواسطة اللغة الإنسانية ، فقد طورت لغات ذات تعليمات محددة لتسهيل عملية التخاطب والاتصال مع الحاسب ، وتتم برجمة الخطوات وإيصالها إلى الحاسب باستخدام إحدى هذه اللغات . ويطلق على الخطوات المترجمة بواسطة تعليمات لغة البرمجة ، والتي تحقق هدفاً محدداً ، اسم البرنامج : (Program) .

ويعكس الرسم التالي العلاقة بين العناصر الثلاثة :



حيث يتم التفاعل والاتصال بين البرنامج والبيانات لإنتاج المعلومات .

الهيكـل الهرمـى للبيانات (Hiearchical Structure of Data) :

حيث إن التعامل مع البيانات يشكل النشاط الرئيسي في تحقيق الهدف من استخدام الحاسب ، كان لابد من وضع قواعد لتسهيل عملية تخزين واسترجاع ومعالجة هذه البيانات .

يطلق اسم «الحقل» على الأوصاف (الخصائص) التابعة للشيء المحدد، فالعمر والاسم يسميان حقلاً .

ويطلق اسم «السجل» على الحقائق (القيم) التابعة لعنصر واحد من الشيء الموصوف . فالحقائق التالية التابعة لطالب معين تعتبر سجلاً :

سالم محمد على الحاسب الآلى ٣,٩٥

ويطلق اسم «الملف» على جميع السجلات التابعة للشيء الموصوف .

وبناء على ذلك يكون الهيكل الهرمى للبيانات على الشكل التالى :

الملف (File)

السجل (Record)

الحقل (Field)

حيث يوجد لكل ملف أكثر من سجل ، ولكل سجل أكثر من حقل . ويتم تخزين البيانات عن طريق الحصول على قيم الحقول المتعددة التابعة لسجل معين ، ومن ثم تخزينها إذا تطلب الأمر في ملف واحد ، ولا تعالج السجلات المتتابعة دون تخزينها في الملف . وإذا تم تخزين الخصائص (القيم) ضمن ملف وأريد استرجاعها ، فمن الممكن استرجاع الملف كاملاً بسجلاته جميعها ، أو يمكن اختيار مجموعة سجلات من الملف ، كاختيار الطلبة المتفوقين الذين تزيد معدلاتهم على ٣,٥٠ ، أو فرز وتصنيف السجلات حسب التخصصات مثلاً ، أو يمكن استرجاع قيم حقول معينة تابعة لسجل وملف محددين ، للحصول على معدل طالب معين ، مثلاً ، من عدة علامات معطاة . وسنعرض

هنا بإيجاز لأساليب استرجاع البيانات من الملفات ، حيث ستتضح كيفية تطبيق هذه الأساليب في الفصل الثالث عشر الخاص بمعالجة الملفات .

أساليب استرجاع البيانات File Access Methods :

تستند أساليب استرجاع البيانات إلى مفهومين أساسيين :

الأول : الاسترجاع المتتالي Sequential ، والثاني : الاسترجاع العشوائي Random .

الاسترجاع المتتالي : إن ترتيب السجلات في ملف معين حسب هذا الأسلوب يسمح باسترجاع هذه السجلات بطريقة متتالية ، وذلك يشبه التسجيل على الأشرطة السمعية والبصرية . وعند محاولة استرجاع هذه البيانات نواجه بحالتين :

١ - استرجاع السجلات كلها في الملف المعين ، كاسترجاع ملف الموظفين لأجل طباعة الشيكات ، أو استرجاع ملف الدارسين لأجل طباعة كشوف العلامات ، وذلك سجلاً سجلاً ودون تخطي أى منها .

٢ - استرجاع سجل معين ، لموظف محدد أو دارس معين ، ولا يتم ذلك إلا بعد استرجاع جميع السجلات السابقة على السجل المطلوب . فمثلاً : إذا كان اسم الدارس الأخير «سالم» ، فيجب استرجاع كل سجل سابق على سجل الدارس والتأكد في كل مرة من أن السجل المسترجع هو المطلوب أم لا ، إلى أن نتوصل إلى استرجاع السجل المطلوب للدارس المعين .

لذلك كان ترتيب السجلات في سجل متتال (متتابع) يستند إلى حقل معين في ذلك السجل مثل رقم الموظف أو اسم الدارس ، وذلك في أغلب الأحيان . وفي بعض الأحيان ، يكون تسلسل السجلات في الملف وفق التسلسل الزمني لوقوع تسجيل المعلومات على كل سجل .

ويتبع هذا الأسلوب في الحالات التي تتطلب استرجاع جميع سجلات الملف بصفة دورية، أسبوعياً، أو شهرياً، أو سنوياً، وذلك مثل تجهيز درجات الدارسين عقب كل دورة أو فصل، أو تحضير الشيكات للموظفين شهرياً، أو تجهيز الفواتير للكهرباء أو الهاتف شهرياً. وكذلك في حالات استرجاع نسبة كبيرة من مجموع السجلات، كتلك التي تتبع منطقة جغرافية معينة أو نوعاً معيناً من الخدمات أو البضائع، بحيث يكون عدد هذه التصنيفات في الملف قليلاً.

الاسترجاع العشوائي : ويسمح هذا الأسلوب باسترجاع أى سجل من السجلات دون الحاجة إلى استرجاع ماسبقه. مع العلم بأن هذا الترتيب يسمح باسترجاع البيانات بطريقة متتالية لنفس الملف. ويتم تحديد ترتيب السجلات واسترجاعها وفق إحدى طريقتين :

١ — استخدام رقم كل سجل لا يكون من الحقول التابعة له والمخزنة في كل سجل، حيث يربط الحاسب — عن طريق البرنامج — بين كل سجل ورقمه. ولاسترجاع السجل المطلوب لابد من تحديد رقمه أولاً، ومن ثم استرجاع الحقول التابعة له.

٢ — استخدام أحد الحقول التابعة للسجل، لتحديد كل واحد منها بطريقة فريدة، كما في ترتيب الملف العشوائي. ومن الأمثلة عن الحقل الذي يمكن استخدامه — والذي يطلق عليه الحقل المفتاح Key Field — اسم الدارس أو الموظف (أو رقمه)، ورقم الفاتورة ورقم الطلبية. ولا يشترط ترتيب السجل تسلسلياً حسب قيمة هذا الحقل كما هو الحال في الملف التتابعى.

ويشرح الشكل التالى الأساليب المذكورة سابقاً، والذي يفترض وجود خمسة سجلات في ملف الدارسين، يحوى كل ملف الحقول التالية: رقم الدارس، اسمه، والمدينة التي جاء منها :

رقم السجل	المدينة	الاسم الأول	الاسم الأخير	رقم الدارس
5	الرياض	حميدان	حامد	12562
4	حائل	يوسف	ثابت	23145
3	أبها	رامي	محمود	56112
2	جدة	سالم	ناصر	35121
1	الدمام	خالد	يحيى	21231

١ - الأسلوب التتابعى : تم ترتيب السجلات وفق الاسم الأخير للدارس ، لذلك فلاسترجاع ملف الدارس رامى محمود ، لابد من استرجاع ملفى الدارسين : حامد وثابت ، قبل أن نتمكن من الوصول إلى الملف المطلوب .

٢ - الأسلوب العشوائى الذى يستند إلى حقل مخزن ضمن السجل : فى هذه الحالة تم تحديد كل سجل باستخدام رقم الدارس غير المكرر ، فإذا أردنا استرجاع نفس السجل ، نخير الحاسب بأننا نرغب فى استرجاع حقول الدارس رقم 56112 ، دون الحاجة إلى استرجاع أية سجلات أخرى قبله .

٣ - الأسلوب العشوائى المستند إلى رقم السجل غير المحفوظ ضمن حقول السجل نفسه : فعند استرجاع نفس السجل ، يتم إعلام الحاسب برقم السجل المطلوب «3» فى مثل هذه الحالة .

عوامل المفاضلة بين أساليب استرجاع البيانات : نوجز هنا بعض العوامل التى تؤثر فى استخدام هذه الأساليب :

١ - التكلفة : إن تكلفة الأجهزة المناسبة لتخزين السجلات بشكل متتابع هى أقل من تكلفة الأجهزة المطلوبة لتخزين السجلات بأسلوب عشوائى .

٢- طبيعة استرجاع البيانات : إن استرجاع السجل كله أو جزء كبير منه في كل مرة يستخدم فيها السجل ، يستدعى استخدام الملف التتابعى . أما إذا كانت الحالة تتطلب استرجاع سجل محدد أو مجموعة قليلة من السجلات (تقريباً ١٠%) في كل مرة يستخدم فيها السجل ، فإن ذلك يتطلب استخدام الأسلوب العشوائى.

٣- الاعتمادية : تعتبر وسائط التخزين العشوائى أكثر اعتمادية من وسائط التخزين المتتالى .

٤- الهدف من التخزين : تستخدم أجهزة التخزين التتابعى فى العادة لحفظ نسخ من السجلات لأغراض المحافظة عليها Back-up Copies ، وكذلك فى حالات التخزين التى تتطلب الحفاظ على معلومات تاريخية لفترة طويلة ، ٣ - ١٠ سنوات مثلاً ، لأغراض حكومية أو أمنية . فى حين تستخدم أجهزة التخزين العشوائى لتخزين المعلومات المستخدمة باستمرار .

٥- السرعة : إن أجهزة الاسترجاع العشوائى (المباشر) أسرع من أجهزة التخزين المتتالى فى الوصول إلى الحقول التابعة للسجلات .

٦- قدرات الاستفهام Inquiry المباشر : تمتلك وسائط التخزين المباشر العشوائى إمكانية الربط المباشر المستمر مع الحاسب ، بحيث تسمح بالاسترجاع اللحظى الآتى للمعلومات والبيانات ، فى حين تخزن فى معظم الحالات ، وسائط التخزين المتتالى بشكل منفصل ومستقل عن الحاسب ، وتحمل فيما بعد عند الحاجة على الأجهزة الخاصة بها .

٧- السعة التخزينية : إن السعة التخزينية لأجهزة الوصول المباشر العشوائى هى أكبر بكثير من تلك السعة الخاصة بأجهزة التخزين المتتالى .

أجهزة التخزين : تنقسم أجهزة التخزين إلى نوعين رئيسيين :

١ - وسائط التخزين التى يتم التخزين المباشر للبيانات عليها ، وذلك كالقرص الرقيق Floppy Diskette أو مجموعة الأقراص Disk Pack ، حيث يستخدم الأول للحاسبات الشخصية الصغيرة ، والأخير للحاسبات الكبيرة . وهذه الوسائط تسمح بالوصول العشوائى للسجلات . أما الأشرطة المغناطيسية Tape التى تشبه تلك الخاصة بأجهزة التسجيل المصغرة Cassette tape أو الأجهزة الكبيرة Tape ، فتستخدم الأولى مع الحاسبات الشخصية ، والثانية مع الحاسبات الكبيرة . وهذه الأشرطة تستخدم عند تطبيق أساليب الاسترجاع المتتالى للسجلات .

٢ - وحدة القراءة والكتابة ، التى تستخدم لاسترجاع البيانات المخزنة على وسائط التخزين ، ويطلق وحدة القرص Disk Unit على الجهاز الخاص بالاسترجاع العشوائى ، ووحدة الشريط Tape Unit على الجهاز الخاص بالاسترجاع المتتالى . انظر الشكل التالى (ص ٤٤) ، الذى يوضح أجزاء أجهزة الاسترجاع العشوائى والمتتالى .

نظرية النظام (Systems Theory) وتطبيقها مع الحاسب الآلى :

تعتبر نظرية النظام إحدى الوسائل المهمة التى تساعد فى فهم كيفية عمل وتفاعل مجموعة الأجزاء والمكونات ، التى تحقق هدفاً معيناً فى كثير من المجالات . هناك النظام المالى والنظام الاقتصادى والنظام التربوى . وبدلاً من الخوض فى تفاصيل أسس وقواعد نظرية النظام ، سيتم تقديم إطار مبسط يفى بالغرض من هذا الجزء من الكتاب .

مقدمة عن نظرية النظام :

يمكن تعريف النظام بأنه «مجموعة من الأجزاء (المكونات) ذات وظائف محددة تتفاعل فيما بينها لتحقيق هدف محدد» ، وبناء على ذلك يمكن تحديد معالم نظرية النظام كالتالى :

نوع الجهاز

المتوسط - الكبير

الشخصي

وسائط التخزين



القرص المتحرك



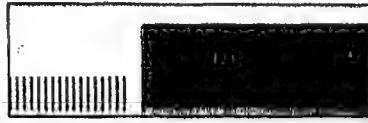
القرص الثابت



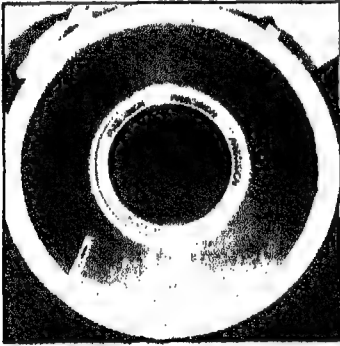
القرص الرقيق



وحدة القرص الثابت من نوع آي. بي. ام I.B.M

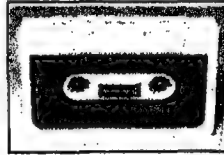


وحدة القرص الرقيق من نوع آي. بي. ام I.B.M

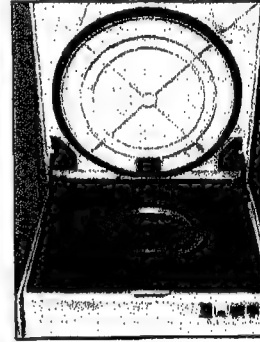


الشريط المغنط

أ - العشوائي.



الشريط

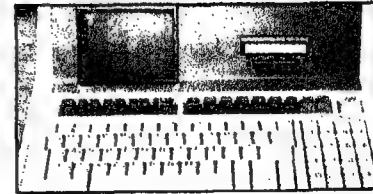


وحدة القرص المتحرك

من نوع تكساس TEXAS



وحدة الشريط المغنط من نوع آي. بي. ام I.B.M



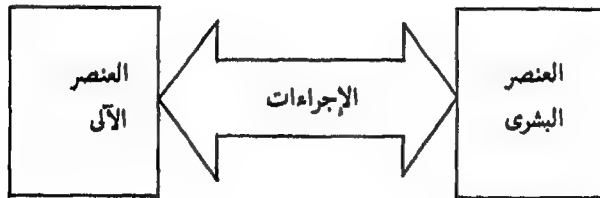
وحدة الشريط من نوع هيويت باكر HP

شكل ١ - ١ : أنواع أجهزة التخزين العشوائي والمتنالي

- ١ - تحديد الهدف من النظام . وهذا الهدف يكون عاماً تلحق به أهداف فرعية تفصيلية ، حسب التعقيد في وظائف الأجزاء وتفاعلها .
- ٢ - تحديد الوظائف التي يجب أن تتم لتحقيق الهدف العام والأهداف الفرعية .
- ٣ - تحديد المكونات (الأجزاء) التي ستقوم بهذه الوظائف . ويلاحظ أن كل جزء يجب أن يقوم بمهمة محددة تؤدي في النهاية إلى تحقيق الهدف العام .
- ٤ - تحديد أسس التفاعل بين هذه الأجزاء حتى يتم تحقيق الهدف العام والأهداف الفرعية ، بطريقة متناسقة ومتجانسة وتمنع أى اضطراب أو خلل في تأدية الوظائف .

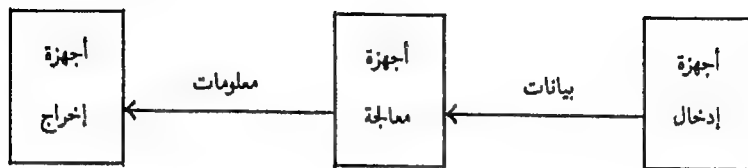
عناصر (مكونات) نظام الحاسب الآلى :

إن الهدف العام من نظام الحاسب الآلى هو تحقيق متطلبات المستخدمين عن طريق الاستخدام الأمثل لأجهزة الحاسب . لذلك فإن نظام الحاسب الآلى يعتمد بشكل رئيسى على عملية الاتصال مع العنصر البشرى (الإنسان) والعنصر الآلى (أجهزة الحاسب) ، ويتم تحديد أسس الاتصال عن طريق الإجراءات التفصيلية . ويعكس الشكل التالى الإطار العام لنظام الحاسب الآلى :



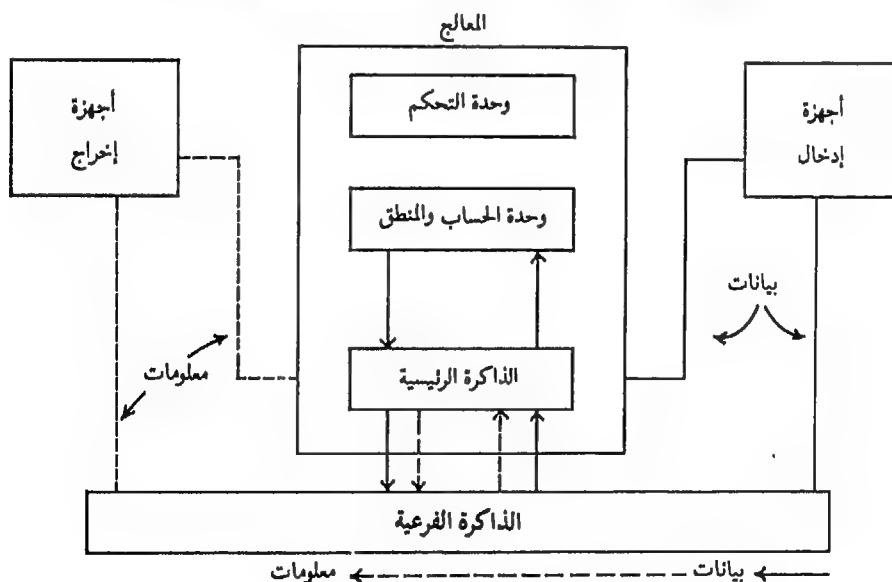
وسيتم التركيز هنا على العنصر الآلى وذلك لأهميته في فهم أسس البرمجة ، في حين ستتم مناقشة العنصرين الآخرين بإيجاز ، وذلك لعلاقتها بموضوع التحليل والتصميم .

العنصر الآلى : لقد تمت مناقشة وظائف الحاسب فى الجزء الأول من هذا الفصل . فالهدف من العنصر الآلى هو معالجة البيانات وإخراجها على شكل معلومات ، لذلك لابد من وجود أجهزة إدخال ومعالجة وإخراج . أما تفاعل هذه الأجزاء فيمكن وصفه بالرسم التالى :



وإذا أضيفت وظائف التخزين والمعالجات الحسابية والمنطقية وضرورة وجود جزء يتحكم فى التفاعل بين هذه الأجزاء جميعاً ، يمكن وصف النظام الفرعى للعنصر الآلى بالشكل التالى . ويطلق تعبير الأجهزة المتعلقة Peripheral devices على أجهزة الإدخال والإخراج والذاكرة الفرعية . هذا وسيتعرض الجزء الأخير من هذا الفصل لأنواع هذه الأجهزة مع الذاكرة الرئيسية .

شكل ١ - ٢ : مكونات الحاسب الآلى وتفاعلها بعضها مع بعض



وبعد تحديد الهدف العام ووظائف ومكونات الأجزاء لنظام عنصر الحاسب الآلى (الأجهزة)، يمكن عرض بعض قواعد التفاعل بين الأجزاء :

١ — لا يمكن لأى جزء أن يقوم بأية مهمة دون أمر مباشر من وحدة التحكم . وهذا يتم داخلياً وليس له علاقة بعملية برمجة التعليمات .

٢ — للحصول على المعلومات لابد من أن تتم المعالجات المطلوبة على البيانات المدخلة . وهذا يتطلب أن تكون البيانات المدخلة صحيحة ، وأن تكون خطوات المعالجة متكاملة وواضحة ، لذلك فإن أى خطأ فى المعلومات يكون سببه إما خطأ فى البيانات ، أو فى ترجمة خطوات المعالجة إلى تعليمات فى البرنامج ، أو عدم وجود البيانات بالكلية ، أو معالجة البيانات التابعة لمعالجة أخرى خاطئة .

٣ — يمكن تخزين البيانات إما فى الذاكرة الرئيسية ، وذلك إذا أريد معالجتها آتياً (لحظياً) ، أو فى الذاكرة الفرعية إذا أمكن الانتظار لحين وجود مكان شاغر، فى الذاكرة الرئيسية . ولما كانت الذاكرة الرئيسية تعتبر الذاكرة النشطة والمستعدة دائماً لاستقبال البيانات ، فمن الأفضل تخفيف العبء عنها باستخدام الذاكرة الفرعية إلا فى الحالات التى تتطلب سرعة ومعالجة آتية لكل سجل على حدة ، كحجوزات الطيران مثلاً . أما إذا كان المطلوب معالجة مجموعة سجلات بعضها مع بعض — عند طباعة الفواتير مثلاً — فيمكن استخدام الذاكرة الفرعية بدلاً من الرئيسية .

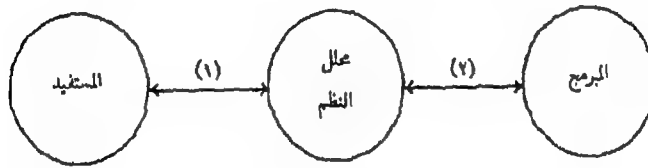
٤ — إنه ، حتى تتم معالجة البيانات ، لابد من وجودها فى الذاكرة الرئيسية . ولذلك فهناك عملية إضافية عند تخزين البيانات فى الذاكرة الفرعية ، يجب الانتباه إليها ، وهى ضرورة نقل البيانات من الذاكرة الفرعية إلى الرئيسية حين الحاجة إلى معالجتها .

٥ - إن وحدة الحساب والمنطق تستقبل وترسل من الذاكرة الرئيسية دون غيرها ، البيانات فقط التى تتطلب معالجة حسابية أو منطقية .

ولابد من التنويه هنا ، وذلك عند التحدث عن الأجهزة (HARDWARE) ، بأن تشغيل هذه الأجهزة والاستفادة من قدرات وخصائص الحاسب يتمان عن طريق البرمجيات (SOFTWARE) بأنواعها المختلفة ، وسيتم التحدث بزيد من التفصيل عن هذا الجزء عند مناقشة عنصر الإجراءات .

العنصر البشرى : يدخل فى هذا العنصر كل من يستخدم الحاسب الآلى ، سواء الموظفون فى مركز الحاسب أو الموظفون فى الجهات المستفيدة من قدرات الحاسب ، كالدائرة المالية ودائرة شؤون الموظفين . وتكون مهمة الموظفين فى مركز الحاسب منصبة على التعرف على رغبات ومتطلبات المستخدمين ، وصياغتها بطريقة يسهل معها تحقيقها بواسطة استخدام الحاسب ، أويديوياً ، أو باستعمال كلتا الوسيلتين .

ويهمنا هنا الاتصال الذى يتم بين ثلاث فئات من المستخدمين ، كما يبينه الشكل التالى ، حيث يدل الرقم على التسلسل المنطقى لهذه الاتصالات :



يقوم محلل النظم ، أثناء وبعد الاتصال مع المستخدم ، بصياغة متطلبات المستخدمين بطريقة تسهل أداء المبرمج لعمله .

ويسلم محلل النظم إلى المبرمج ما له علاقة بكتابة البرامج وفق إحدى لغات البرمجة ، وتتضمن التالى :

١ — تحديد المدخلات للبرنامج .

٢ — وصف الإجراءات الحسابية والمنطقية بإحدى الوسائل المتاحة لمحلل النظم .

٣ — تحديد المخرجات وأسلوب وشكل (عرض) المعلومات .

هذا إذا كانت البرامج المطورة جزءاً من نظام متكامل يخدم فئات متعددة من المستخدمين . أما إذا كان البرنامج ذا هدف محدد ليس له علاقة ببرامج أخرى ، فقد يقوم المبرمج بتحديد البيانات المدخلة والمعالجات والمخرجات والمعلومات . هذا حال التمارين في أى كتاب في البرمجة إلا في بعض الأمثلة المعقدة ، وهى قليلة . وفى هذا الكتاب ، سيتم تحديد أسلوب وشكل عرض المعلومات لبعض التمارين ، ويطلب ذلك من المبرمج فى البعض الآخر؛ وذلك لإعطائه فرصة الممارسة التى ستفيده عند كتابة تطبيقات خاصة به ، أو عند عمله فى جهة قد تتطلب الاتصال مع المستخدمين أو محلل النظم .

عنصر الإجراءات : تتضمن الإجراءات وصفاً للخطوات والعمليات والأنشطة التى تصاحب استغلال قدرات الحاسب من قبل المستخدمين .

وتنقسم الإجراءات إلى قسمين : تلك التى تخص المستخدم ، وتلك التى تخص الموظفين فى مركز الحاسب . وسيتم إعطاء صورة إجمالية عن هذه الإجراءات مع التمرس لأهميتها .

وإجراءات المستخدمين لها علاقة باستخدام الحاسب وتجميع البيانات وتسلم وتوزيع المعلومات ، وأهمها هو دليل المستخدم الخاص باستخدام البرامج التى طورت لتحقيق متطلبات المستخدمين ، مثل هؤلاء الذين يعملون فى الدوائر المالية والإدارية وشؤون الموظفين .

وتنقسم إجراءات المستخدمين فى مركز الحاسب الآلى إلى تلك التى لها علاقة بالنظم التشغيلية (برمجيات النظم Systems Software) وهى غالباً ما تكون مكتوبة من

قبل الشركات المصنعة للحاسب . وتلك التى لها علاقة بالنظم التطبيقية (برمجيات التطبيقات Applications Software) ، وهى التى طورت من قبل المحللين والمبرمجين ، سواء داخل المؤسسة أو عن طريق الاستئجار أو الشراء . وموضوع هذا الكتاب له علاقة بالقسم الأخير .

وتتبع عملية توثيق إجراءات النظم التطبيقية ، نشاطات ووسائل متعددة تؤدى إلى تحضير أدلة وتقارير ذات أهداف محددة . فمن الأدلة ماله علاقة بوصف إجراءات التحليل والبرمجة ، أو بوصف إجراءات التشغيل لتطبيق معين على الحاسب . ومن التقارير ما يقدم على مراحل مختلفة إلى المستخدمين ، وذلك لإطلاعهم على التقدم فى تطوير تطبيق ما ، وإعطائهم معلومات تساعد فى اتخاذ قرارات حول جدوى التطبيق المطور .

وما يهمنا فى هذا الكتاب هو التوثيق المبسط للبرامج المكتوبة من تخصيص البيانات المدخلة ، ووصف لخطوات المعالجة باللغة العربية وتخصيص المخرجات وأشكال عرض المعلومات ، وذلك للتدليل على أهمية التوثيق للبرامج .

أقسام الحاسبات الآلية حسب حجمها :

كان اتجاه حجم الحاسبات خلال التطور التاريخى لها هو من الكبير إلى الصغير ، وهذا بخلاف تصغير حجم الحاسب التابع لفئة معينة . وقد نجم ذلك كله عن الأساليب الحديثة فى تصنيع أجزاء الحاسبات الآلية ، وبشكل خاص جزء المعالج ، وعن استعمال التيار الكهربائى فى تصميم الحاسب ، عوضاً عن الأساليب الإلكتروميكانيكية أو الإلكترومغناطيسية . ولا يهمنا هنا التفصيل فى هذا الموضوع ، ولكن يهمنا فقط التنبيه إلى أن جميع أعمال الحاسب الآلى الحالى تتم عن طريق إرسال تيارات كهربائية عبر أسلاك دون أن يتم تحريك أى جزء أو قطعة منها .

ومن الممكن تقسيم الحاسبات حسب أحجامها إلى الفئات التالية مع التعرض لأهم العوامل التى تتطلب استخدامها :

١ - الحاسبات ذات الحجم الضخم : (Super-Computers) وتستخدم هذه الحاسبات فى التطبيقات التى تتطلب سرعة ودقة متناهيتين فى معالجة أرقام كبيرة، مثل الفضاء واكتشاف وتصنيع البترول .

٢ - الحاسبات ذات الحجم الكبير : (Main-Frame) وتستخدم هذه الحاسبات فى الشركات الكبيرة التى تتطلب تخزين البيانات الكثيرة مركزياً مع تحقيق سرعة ودقة مناسبة، مثل المؤسسات المالية الكبيرة وبعض تطبيقات المؤسسات الحكومية الكبيرة، مثل الداخلية وديوان الخدمة المدنية .

٣ - الحاسبات ذات الحجم الوسط (Mini-Computers) : تستخدم هذه الحاسبات فى الشركات متوسطة الحجم والتى تتطلب مركزية التخزين للمعلومات، أو فى الشركات الكبيرة حيث يتم تخزين لا مركزى لجزء من البيانات والتى تتم معالجتها محلياً، أو عند استعمالها لتطبيق خاص مثل التصنيع والتصميم بمساعدة الحاسب (CADQAM) لتطبيقات المعاهد والمؤسسات التعليمية والتدريبية .

٤ - الحاسبات ذات الحجم الصغير (small Computers) : تستخدم هذه الحاسبات فى الشركات الصغيرة، والتى لا تحتاج إلى طاقة تخزينية عالية حيث كمية البيانات محدودة إلى حد ما، ولا إلى التنوع والكثرة فى أجهزة الإدخال والإخراج، أو فى الشركات الأخرى التى لا تحتاج إلى مركزية فى تخزين ومعالجة البيانات .

٥ - الحاسبات الشخصية ذات الحجم المصغر (Personal Microcomputers) وهذه الحاسبات إما أن تكون عادية ثابتة توضع فوق المكتب، أو متنقلة يمكن حملها كحقيبة أثناء السفر، وتستخدم للأغراض الشخصية سواء فى الأعمال الخاصة

بنشاطات الشركات التابعة لشخص واحد أو لأشخاص قليلين، أو في أعمال المديرين التي تتطلب معالجات ذات صفة خاصة نابعة من طبيعة وظائفهم كالقيام بالتحليلات الإحصائية، أو متابعة النشاطات المنزلية لأفراد الأسرة، كمراقبة المصروفات والإيرادات .

وتحدد هذه الحاسبات عدد المستخدمين بمستخدم واحد في معظم الأحيان (Single User)، مع أن التقدم في تصنيع أجهزة المعالجة أدى إلى إتاحة استخدامها من قبل مستخدمين متعددين (Multi-Users)، مع أن عدد المستخدمين قد لا يصل إلى ذلك العدد المتاح على الحاسبات ذات الحجم الصغير .

وقد تستخدم هذه الحاسبات إما مستقلة بذاتها أو متصلة مع حاسب من النوع ذي الحجم الوسط أو الكبير، وتتمتع هذه الحاسبات بسهولة توصيلها إلى مصدر للتيار الكهربائي وقلة صيانتها .

٦ - الحاسبات المنزلية ذات الحجم الصغير (Home Micro-Computers) وهي شبيهة بالحاسبات الشخصية إلا أن قدراتها أقل منها . وهي غالباً ما تستعمل في الترفيه والتسلية عن طريق استخدام الألعاب، مع إمكانية استخدامها لأغراض متابعة مصروفات وإيرادات الأسرة، وهي من ذات المستخدم الواحد (Single-User) . ومن الممكن تبيان الفروق بين هذه الأنواع من الحاسبات من خلال مناقشة بعض الخصائص :

١ - الحجم : تتراوح أحجام الحاسبات من تلك التي تشبه حقيبة سفر إلى تلك التي تملأ حجرة كبيرة الحجم، هذا دون اعتبار لأجهزة الإدخال والإخراج التي قد تكون موزعة في أماكن متباعدة وتأخذ حيزاً كبيراً .

٢ - حجم الذاكرة الرئيسية : تتراوح حجم الذاكرة الرئيسية بين ٤ آلاف حرف (ك) ٣٢ مليون حرف (٣٢ م ح) .

٣ - حجم الذاكرة الفرعية : كما مر ذكره، تقسم الذاكرة الفرعية إلى ذاكرة وصول متتالٍ وذاكرة وصول عشوائي . وكذلك تختلف الوسائط المستخدمة لكل منهما حسب حجم الحاسب، فالحاسبات الشخصية المصغرة من جهة، والحاسبات الأخرى من جهة ثانية .

وتختلف سعة التخزين المتتالي - الأشرطة - حسب طولها .

أما سعة التخزين العشوائي - الأقراص - فتتفاوت حسب الوسائط المستخدمة . فالقرص الرقيق التقليدي تأتي سعته إما ٣٦٠ ألف حرف أو ١,٢ مليون حرف، أما القرص الصلب المصغر، والذي يستخدم مع جهاز IBM PS/2، فيمكن أن يكون إما ٧٢٠ ألف حرف أو ١,٤٤ مليون حرف . هذا فيما يتعلق بالحاسبات الشخصية . أما في الحاسبات الكبيرة، فتتفاوت سعة وسائط التخزين العشوائي لمجموعة الأقراص .

٤ - عدد وحدات الإدخال والإخراج : تعتبر الطرفيات النهائية المكونة من لوحة الأزرار والشاشة التلفازية أحد أجهزة الإدخال والإخراج الرئيسية على التوالي لكل الحاسبات . في حين تعتبر الطابعات، بأنواعها المختلفة، إحدى الوسائل الشائعة للحصول على معلومات مكتوبة . وكذلك من الممكن كما مر ذكره، استخدام الذاكرة الفرعية كوسيلة إخراج وإدخال أيضا، وذلك إذا ما تم تخزين البيانات المخرجة عليها، ومن ثم أدخلت لمعالجتها .

ويتراوح عدد الوحدات المتعلقة بالحاسب (Peripheral Devices) بين وحدتين وما يزيد على ٥٠٠ وحدة . ففي الحاسبات الصغيرة ذات المستخدم الواحد يمكن إيصال طابعة ومسجل بالإضافة إلى لوحة الأزرار والشاشة .

أما في الحاسبات الكبيرة، فيمكن إيصال عدد كبير من النهائيات الطرفية ووحدات إدخال الأشرطة ووحدات إدخال الأقراص والطابعات، وذلك حسب الحاجة إلى كل نوع منها .

٥ - السرعة : تقاس سرعة الحاسب في القيام بالعمليات الحسابية والمنطقية بـ (١/ ١٠٠٠) من الثانية . في حين تقاس سرعة استجابة الحاسب لطلب المستخدم ، من لوحة الأزرار إلى الشاشة مثلاً ، بالثواني . وكلما زاد حجم الحاسب زادت السرعة في إنجاز أعماله .

٦ - أنواع وتعدد لغات البرمجة : كلما زاد حجم الحاسب زاد معه عدد وأنواع لغات البرمجة . ففي الحاسبات ذات الحجم المصغر (Micro-Computers) ، تعتبر لغة بيسك - اللغة المشروحة في هذا الكتاب - هي اللغة الرئيسية للحاسب ، حيث تستخدم للاتصال مع الذاكرة الرئيسية حال توصيل الحاسب إلى مصدر تيار كهربائي . ومن الممكن تغيير هذه اللغة إلى إحدى لغات قليلة أخرى ، مثل : كوبول ، وفورتران ، وباسكال . أما في الحاسبات الأكبر حجماً ، فإن عدد وأنواع اللغات تكون كثيرة ، بالإضافة إلى إمكانية التخاطب والاتصال مع الحاسب باستخدام أكثر من لغة في نفس الوقت .

تمارين

- ١ — عدد خمسة من الأوصاف المتعلقة بالدارس .
- ٢ — مؤسسة استشارية تقدم خدمات للزبائن عن طريق تقارير عن طبيعة الصناعات المختلفة في الشرق الأوسط . حدد اثنين من الأشياء الموصوفة في هذه المؤسسة . ومن ثم حدد خمسة من الأوصاف لكليهما .
- ٣ — أعط خمس مجموعات من الحقائق (القيم) للأوصاف التالية المتعلقة بالشئ الموصوف .
الشئ الموصوف : الزبون .
الأوصاف : رقم الزبون ، اسم الزبون ، عنوانه ، الرصيد المستحق ، تاريخ آخر دفعة ، تاريخ آخر طلبية .
- ٤ — ماهو المقصود بالتعابير التالية :
نظام ، أجهزة متعلقة ، برمجيات ، مبرمج ، محلل نظم ، مستفيدين ؟
- ٥ — اشرح باختصار وظائف الحاسب الرئيسية والفرعية .
- ٦ — ماهو الفرق بين البيانات والمعلومات والتعليمات ؟
- ٧ — أعط مثالاً عن شئ موصوف مع تحديد ثلاثة أوصاف وإعطاء خمس قيم لهذه الأوصاف .
- ٨ — ماهو الهيكل الهرمي للبيانات ، مع إعطاء مثال على ذلك ؟
- ٩ — اشرح باختصار العناصر الثلاثة لاستخدام نظام الحاسب الآلى ، وما هي العلاقة بينها .

١٠ - اشرح باختصار مكونات الحاسب الآلى وتفاعلها بعضها مع بعض بواسطة

الرسم ؟

١١ - قارن بين أنواع الحاسبات حسب الخصائص التالية :

الحجم ، عدد المستخدمين ، نوع الذاكرة الفرعية ، عدد وحدات الإدخال والإخراج ، السرعة .

مقدمة عن البرمجة التركيبية الحديثة

(Structured Programming)

كان الاهتمام الأكبر في بداية الخمسينيات ينصب على نشاط البرمجة عند تطوير نظم التطبيقات . إذ أن نجاح المحاولات الأولى في تحقيق الفائدة في مجال السرعة والدقة في معالجة البيانات الكثيرة، أدى إلى التسارع في كتابة البرامج في المجالات المختلفة الأخرى، وذلك حتى تعم الفائدة . ولم يكن هناك أى سبب يدعو إلى التفكير في وضع أسس وقواعد لأساليب البرمجة .

أساليب البرمجة التقليدية :

في هذا الجو المتفائل، تم تطوير كثير من البرامج باستخدام الأساليب التقليدية التى لا تخضع لأى أسس أو قواعد . ونظراً لكون البرمجة فناً وليست علماً ذا قواعد ثابتة، كانت البرامج المطورة في ظل هذه الأساليب تتأثر إلى حد كبير بالخبرات الشخصية والفروق الفردية والميول الخاصة .

ومن خصائص أساليب البرمجة التقليدية :

- ١ — عدم وجود أية أسس وقواعد تتبع من قبل المبرمجين .
- ٢ — نظرة المبرمج إلى البرنامج وكأنه إنتاج خاص به، وأن تلازم المبرمج والبرنامج سيكون دون انقطاع .
- ٣ — عدم الأخذ بعين الاعتبار بأن البرامج المطورة ستخضع للتعديل فيما بعد .

٤ - كتابة البرامج دون تحليل مسبق ، مما أدى إلى عدم ترابط البرامج المختلفة المطورة على فترة من الزمن .

٥ - عدم وجود تعليمات في لغة البرامج للتحكم في الانتقال المبني على نتيجة المقارنات المنطقية بين قيمتي حقلين ، أو أن استخدامها حال وجودها لم يكن ينفع لأى ترتيب واضح . لذلك كان المبرمج يقوم بالانتقال إلى أى تعليمة في البرنامج سواء كانت سابقة أو لاحقة .

٦ - عدم تقسيم البرامج الكبيرة إلى أجزاء مترابطة بعضها مع بعض ، وقد اكتنفت البرامج التى طورت بهذه الأساليب صعوبات عديدة :

١ - صعوبة إمكانية تنقل البرامج من مؤسسة إلى أخرى إلا بصرف جهد ووقت كبيرين .

٢ - صعوبة إجراء تعديلات على هذه البرامج .

٣ - صعوبة فهم البرامج من قبل الأفراد الآخرين .

٤ - صعوبة التحقق من صحة المعلومات المخرجة .

٥ - صعوبة متابعة تأثير التغيير الذى تم فى جزء من البرنامج على الأجزاء الأخرى .

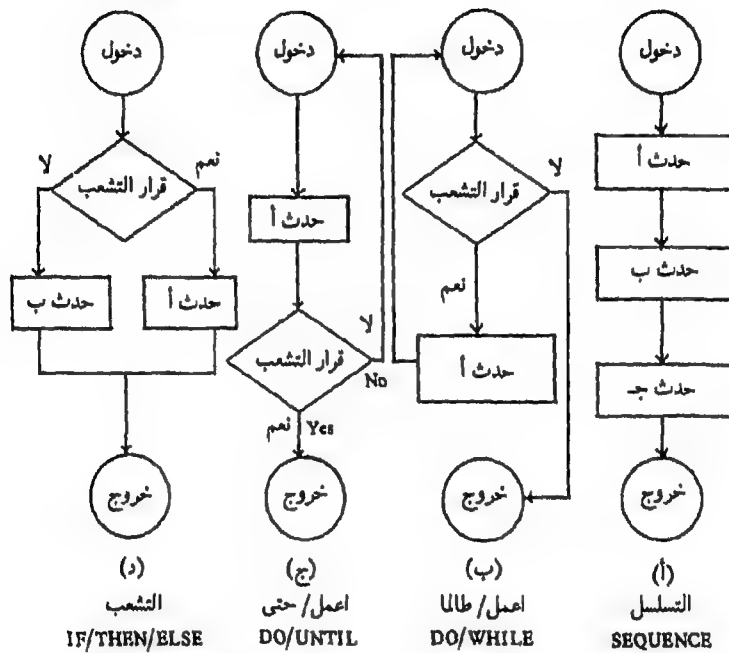
٦ - صعوبة معرفة إذا ما كان الخطأ الذى وقع فى جزء من البرنامج قد سبب وقوع أخطاء فى الأجزاء الأخرى .

وللتقليل من هذه الصعوبات ظهرت جهود، بدأها الأستاذ دجكسترا (Edsger W. Dijkstra) العالم النمساوى ترمى إلى وضع أسس وقواعد للبرمجة الحديثة أطلق عليها «البرمجة التركيبية Structured Programming» .

أساليب البرمجة التركيبية الحديثة :

تحدد البرمجة التركيبية (ب ت) أربعة أنماط رئيسية، لا تباعها عند ترجمة الخطوات الخاصة بتحقيق الهدف من البرنامج إلى تعليمات إحدى لغات البرمجة . ويمكن توضيح هذه الأنماط الأساسية بالأشكال التالية :

شكل (٢ - ١) الأنماط الأربعة الأساسية للبرمجة التركيبية



وقبل مناقشة هذه الأساليب، لابد من توضيح الرموز المستعملة في الرسومات .

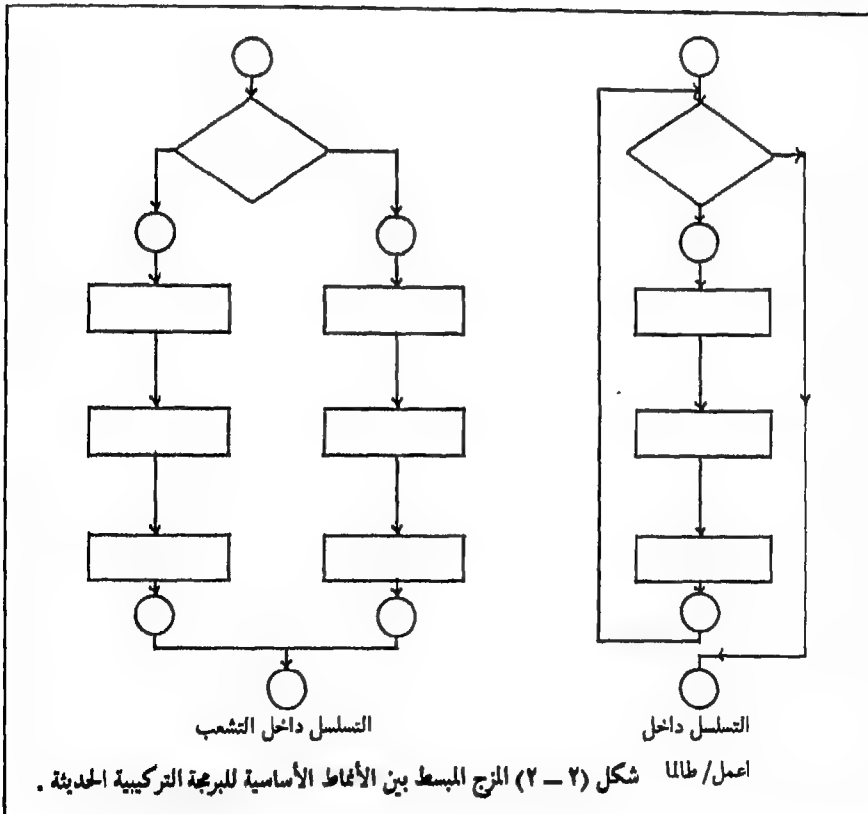
○ تعبر عن بداية ونهاية النمط المحدد .

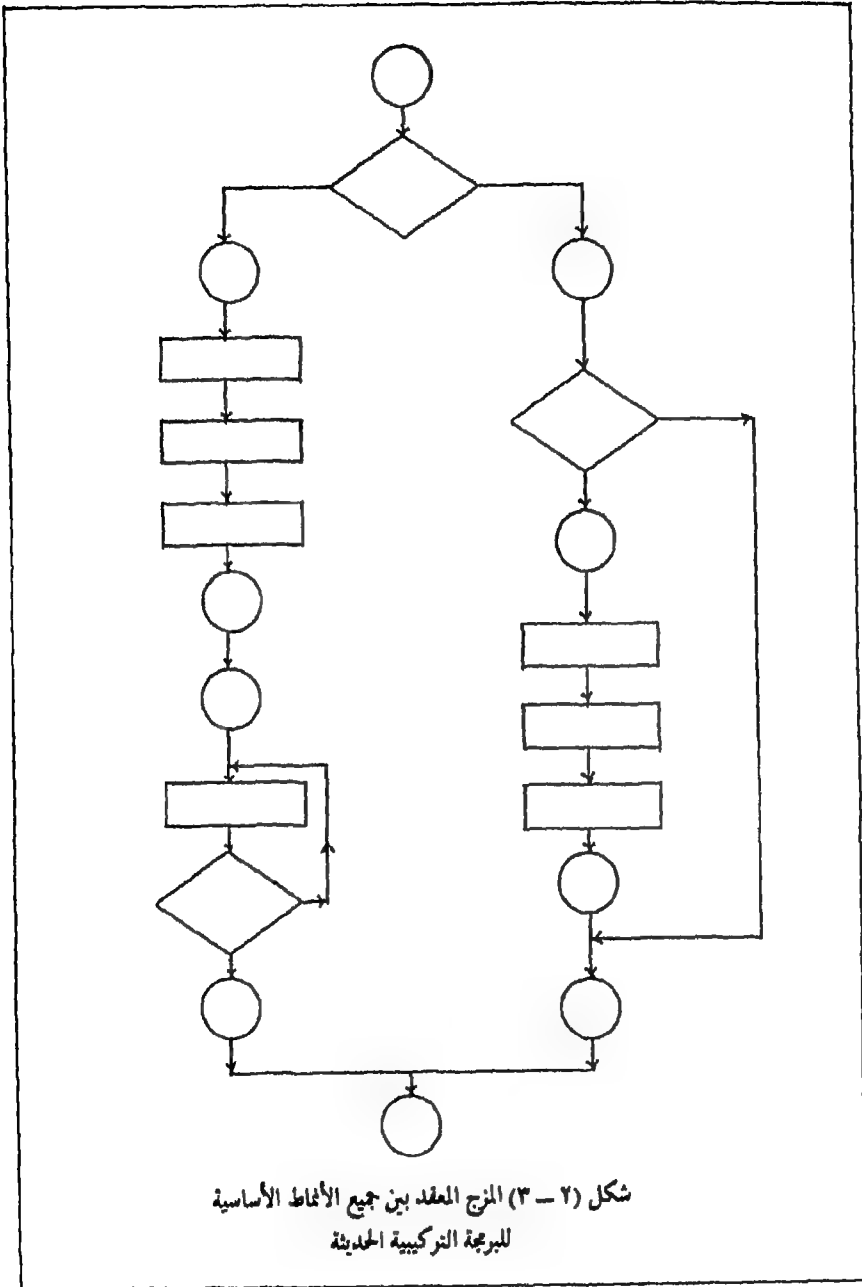
□ تعبر عن معالجة حسابية بإحدى لغات البرمجة .

◇ تعبر عن اتخاذ قرار باختيار أحد المسارين استناداً إلى نتيجة المقارنة المنطقية بين قيمتي حقلين من البيانات .

▱ تعبر عن عمليات الإدخال و/ أو الإخراج .

وفي الشكل (٢ - ١) يوضح الرسم (أ) التعليمات التي ستنفذ دون إجراء أية مقارنات تؤدي إلى التشعب، أي أن التعليمات ستنفذ في تسلسل ثابت .
بينما الرسم (ب) يوضح أن نتيجة المقارنة المنطقية تحدد إذا ما كانت التعليمة ستنفذ أم لا .
بينما الرسم (ج) يوضح أن التعليمة ستنفذ أولاً مع استمرار تنفيذها استناداً إلى نتيجة المقارنة المنطقية .
بينما الرسم (د) يوضح اختيار تنفيذ إحدى التعليمتين بناء على نتيجة المقارنة المنطقية .
ومن الممكن أن تتطلب خطوات المعالجة المزج بين نمطين أو أكثر . وتعكس الأشكال التالية أمثلة من هذا النوع .





ويلاحظ من الأشكال السابقة، أن استخدام أساليب البرمجة الحديثة فن يتقن عن طريق الممارسة، ويتأثر بالإبداع الشخصي للمبرمج .

ولاستخدام هذه الأساليب تتبع القواعد التالية :

- ١ — تحديد بداية ونهاية كل نمط كجزء مستقل بذاته .
- ٢ — عدم تنفيذ تعليمة منفصلة داخل أى نمط محدد، وإنما يكون التنفيذ من بداية النمط إلى نهايته .
- ٣ — عدم الانتقال إلى تعليمة داخل النمط، وإنما يكون الانتقال إلى بداية النمط أو نهايته حسب الحاجة .
- ٤ — عدم الانتقال من أية تعليمة داخل النمط المحدد إلى أية تعليمة أخرى خارجها، وإنما يكون الانتقال إلى نهاية النمط فقط .

خصائص وفوائد البرمجة التركيبية :

ينتج عن استخدام أساليب البرمجة التركيبية برامج ذات خصائص معينة، وفوائد بنيت عليها، يمكن تلخيصها كالتالى :

الخصائص :

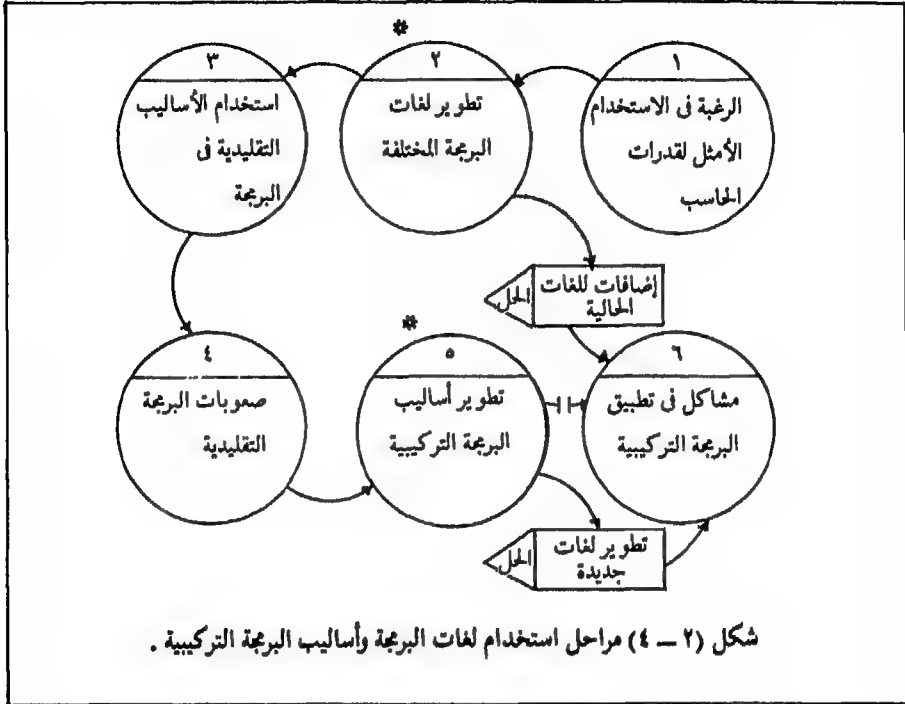
- ١ — سهولة التحقق من صحة المعلومات المخرجة من البرنامج .
- ٢ — الوضوح وسهولة قراءة ومتابعة خطوات البرنامج، وبشكل خاص من قبل الآخرين .

الفوائد :

- ١ — سهولة إجراء التعديلات على البرنامج .
- ٢ — زيادة إنتاجية المبرمجين بنسبة ١,٥ إلى ٢ .

المشاكل التى تعترض تطبيق البرمجة التركيبية :

نجد أن المشاكل التى اعترضت تطبيق البرمجة الهيكلية ، جاءت نتيجة لمراحل التسلسل التاريخى التى مرت بها عملية التخاطب مع الحاسب ، و يعكس الرسم التالى هذه المراحل :



يلاحظ أن السبب الرئيسى فى ظهور المشاكل هو عدم التوافق بين تعليمات لغات البرمجة وأساليب البرمجة التركيبية ، وذلك لكون اللغات قد طورت قبل الأساليب ، كما هو مبين فى تسلسل الدوائر المؤشر عليها بـ « * » .

لذلك كانت حلقة الوصل المفقودة بين تطوير الإطار النظرى لأساليب البرمجة التركيبية ، وتطبيقها فى كل الظروف التى صاحبت تطويرها ، كما هو مبين فى شكل (٢ - ٤) الخط المقطوع بين الدائرتين رقم ٥ و ٦ .

وقد أخذت الجهود الرامية إلى معالجة مشاكل تطبيق البرمجة التركيبية (ب ت) اتجاهين :

١ - تعديل اللغات الحالية عن طريق إضافة تعليمات وخصائص تسهل تطبيق الـ (ب ت) .

٢ - تطوير برامج جديدة ذات تعليمات وخصائص توافق أسس وقواعد البرمجة التركيبية .

وقد ركزت الجهود على الحل الأول ؛ وذلك لسببين :

(١) للوقت الكبير المستثمر في البرامج الحالية التي طورت باللغات الموجودة ،

(٢) لتوفر المبرمجين ذوى الخبرة في استخدام هذه اللغات .

وقد واجه تنفيذ هذين الحلين الصعوبات التالية :

١ - صعوبة تغيير ما اعتاد عليه المبرمجون من أساليب البرمجة التقليدية مع توفر الرغبة في التغيير .

٢ - صعوبة التغلب على خاصية الطبيعة البشرية في مقاومة كل جديد ، سواء في استخدام أساليب البرمجة التركيبية أو في استخدام اللغات الجديدة .

٣ - صعوبة التزامن بين الوقتين اللازمين لاكتساب الخبرة في اللغات الجديدة ، ولتلبية متطلبات المستفيدين ؛ إذ يتخلف الأول عن الثانى في معظم الحالات .

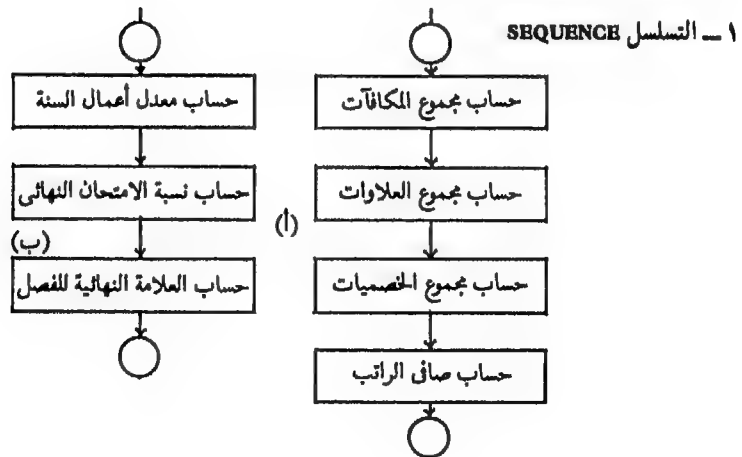
٤ - صعوبة التوفيق بين الأساليب التقليدية في البرمجة وأساليب البرمجة التركيبية عند اجتماعهما في برنامج واحد . وينتج هذا عند اتباع أساليب البرمجة التركيبية في التعديلات التي تتم على البرامج الحالية . وقد يؤدى هذا إلى إعادة كتابتها وفق أساليب البرمجة التركيبية .

للتعرف على مدى قابلية أى من لغات البرمجة ، لابد من تحديد بعض الخصائص التي يجب أن تتصف بها :

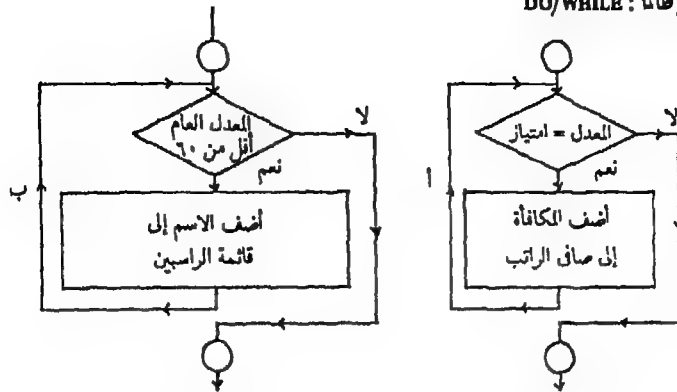
- ١ — تحديد نقطتي البداية والنهاية للنمط .
 - ٢ — تنفيذ مجموعة من التعليمات المترابطة بتعليلة واحدة .
 - ٣ — اختلاف مواقع بدء كتابة التعليمات في الأسطر المترابطة ، وأن التحكم في مساحة الهوامش بهذا الأسلوب يؤدي إلى زيادة الوضوح في البرنامج .
 - ٤ — تحديد بداية ونهاية تعليمات المقارنة المنطقية ، وخاصة المعقدة منها والتي تتطلب تسلسلاً متداخلاً للعديد منها ، وبطريقة منطقية .
- ومن أمثلة اللغات التي قام الباحثون بتعديلات وإضافات عليها لتتكيف مع أساليب البرمجة التركيبية : الكوبول ، والفورتران ، والبيسك (لغة هذا الكتاب) ، في حين تم تطوير لغة الباسكال ، مثلاً كلغة متكاملة وفق أساليب البرمجة التركيبية . وقد لا توجد اللغات المعدلة أو الحديثة في كل المؤسسات وعلى كل الحاسبات ، لذلك على المبرمج أن يحاول قدر استطاعته التقيد بأساليب البرمجة التركيبية ضمن اللغات المتاحة ، وذلك لتحقيق أكبر قدر ممكن من الاستفادة .

أمثلة مبسطة عن أساليب البرمجة التركيبية :

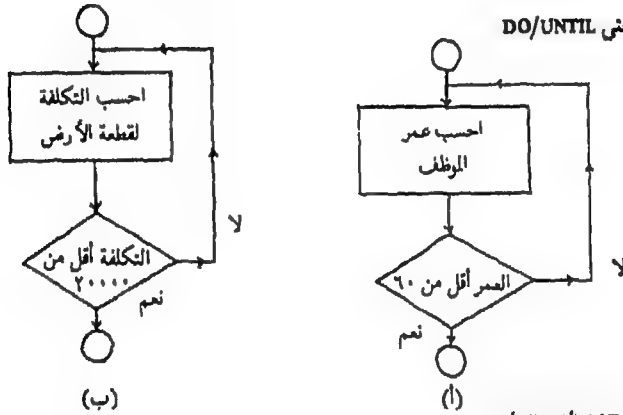
إليك بعض الأمثلة عن الأنماط المختلفة للبرمجة التركيبية :



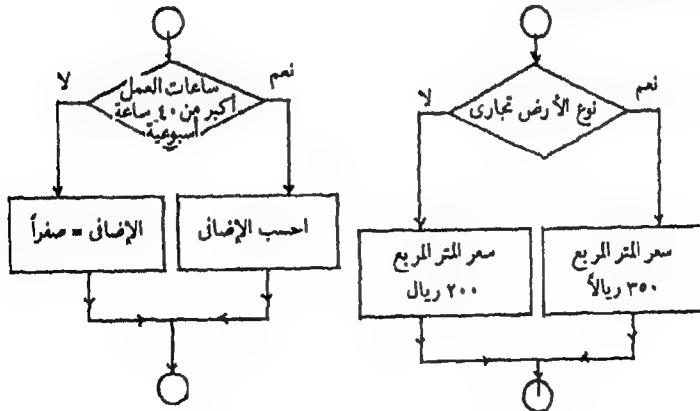
٢ - اعمل / طالما : DO/WHILE



٣ - اعمل / حتى : DO/UNTIL

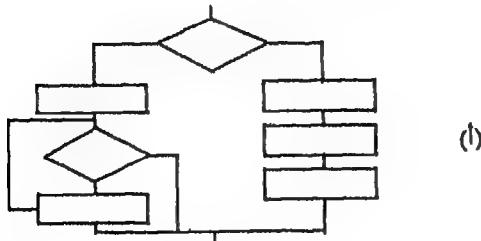


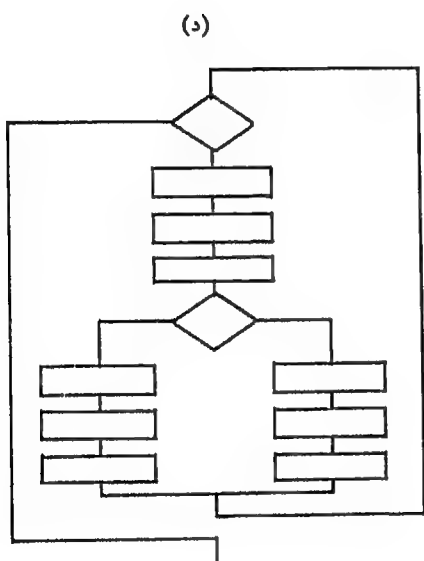
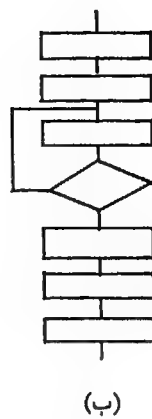
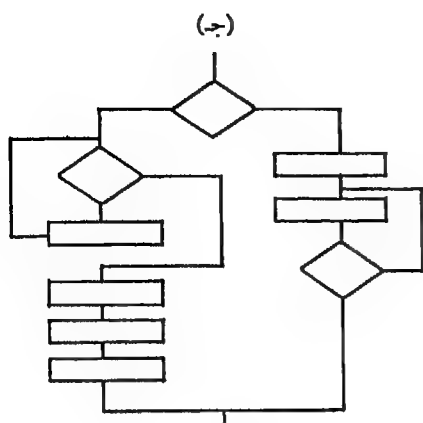
٤ - التشعب : IF/THEN/ELSE



تمارين

- ١ - اختر النمط المناسب للمسائل التالية، ومن ثم ارسم الشكل الخاص به .
 - أ - تاجر يبيع نوعين من السجاد : العادى والفاخر .
سعر السجاد العادى ٢٠ ريالاً للقدم المربع ، وللسجاد الفاخر ٣٥ ريالاً للقدم المربع .
 - ب - متعهد بناء يستمر بحساب مساحة مجموعة من البناءات إلا إذا زادت هذه المساحة على ٥٠٠ متر مربع .
 - ج - يتم حساب مكافأة لكل عامل زادت كمية إنتاجه على ٣٠٠ قطعة .
 - د - للتوصل إلى تكلفة صناعة لتر واحد من «البوظة» تم حساب التكاليف التالية : الحليب، السكر، الزبدة، النكهة .
 - هـ - يعطى الزبون خصم ١٠٪ إذا زادت مشترياته الشهرية على ٥٠٠ ريال .
 - و - إذا كانت شهادة الطالب هى الثانوية، يضاف اسمه إلى قائمة الطلبة المؤهلين للالتحاق ببرنامج الدبلوم العادى، وإذا كان من حملة شهادة البكالوريوس يضاف اسمه إلى قائمة برنامج الدبلوم العالى .
 - ز - يتم التوصل إلى صافى الربح بعد معرفة مجموع الإيرادات ومجموع المصروفات .
- ٢ - حدد نوع النمط من أساليب البرمجة التركيبية فى الأشكال المركبة التالية، وأضف الدوائر الصغيرة عند بداية ونهاية كل خط :





مقدمة عن التجزئة الوظيفية الهيكلية (الهرمية)

(Top-Down Functional Decomposition)

تعتبر التجزئة الوظيفية الهرمية الأساس الثاني الذى تستند إليه الأساليب الحديثة فى البرمجة ، وذلك بالإضافة إلى البرمجة التركيبية التى تم شرحها فى الفصل السابق . ولعلنا نفرق هنا بين أسلوب البرمجة الذى يقضى بتقسيم البرنامج إلى أجزاء دون أن يكون هناك أى تخطيط سابق يرمى إلى ربط هذه الأجزاء بطريقة منطقية قبل تنفيذ عملية البرمجة الفعلية لهذه الأجزاء ، وبين الأسلوب الذى يقضى بالدراسة المسبقة لمتطلبات المستفيدين من أجل الوصول إلى تعريف للوظائف الأساسية والفرعية ، ومن ثم يتم تحديد الأجزاء التى تحقق هذه الوظائف وكيفية ترابطها بعضها مع بعض وفق قواعد متكاملة . هذه القواعد تتبع نمطاً معيناً يأخذ شكل الشجرة المعكوسة : أصلها فى السماء وفروعها متدنية إلى أسفل ، أو ما يطلق عليه «التركيب الهرمى» .

من الممكن استخدام أسس البرمجة التركيبية دون أن يصاحب ذلك اتباع لأساليب التركيب الهرمى ، مع أن البرمجة التركيبية بطبيعتها قد تفرض مستوى معيناً محدوداً من التجزئة البرمجية ، ويتم ذلك عن طريق تحديد بدايات ونهايات كل جزء يتبع نمطاً معيناً من أنماط البرمجة التركيبية . والعكس صحيح أيضاً ، فقد تتبع أساليب التجزئة الوظيفية الهرمية دون أن تتبع معها أساليب البرمجة التركيبية . وأكثر ما ينطبق ذلك على وضع تستخدم فيه إحدى لغات البرمجة التى لا تدعم أساليب البرمجة التركيبية بدرجة كبيرة . ولعل الوضع الأمثل هو فى أن تتكامل أساليب البرمجة الحديثة فيما بينها ، حتى يتم تطوير البرامج وفق أسس كلا الأسلوبين ، ومن ثم تتحقق الاستفادة المرجوة من استخدامهما .

أسس التجزئة الوظيفية الهرمية

إن للتجزئة الوظيفية الهرمية أساسين رئيسيين : معرفة الوظائف المزمع القيام بها ، وترتيب هذه الوظائف بواسطة هيكل هرمي .

الأساس الأول : معرفة الوظائف :

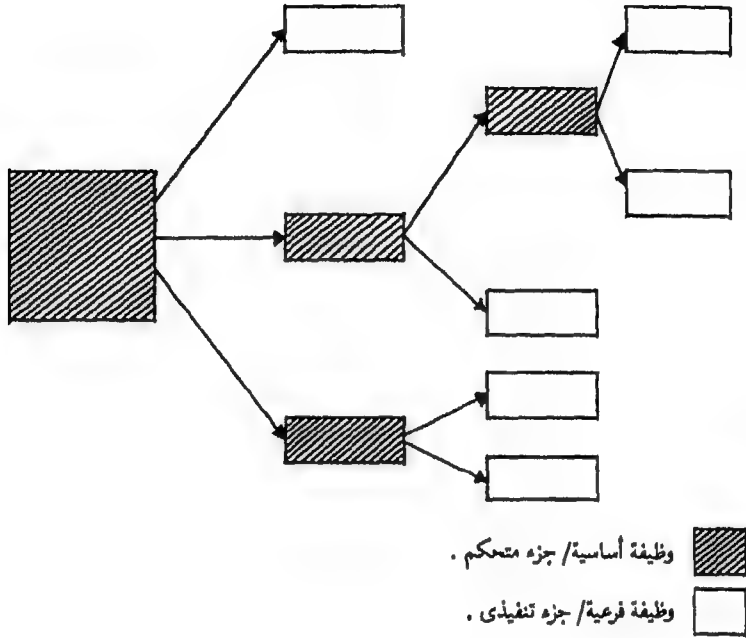
يبدو للوهلة الأولى أن الوصول إلى تحديد للوظائف أمر سهل . قد يكون هذا صحيحاً لدرجة ما إذا كانت متطلبات المستفيدين بسيطة وواضحة ولكن في معظم الأحيان ، لا يكون ذلك واقعياً ، إذ كثيراً ما تكون الوظائف كثيرة ومتشعبة وغير واضحة المعالم . ويحتوى هذا الكتاب على أمثلة وتمارين ذات مستويات متعددة ، ولما كان الهدف الرئيسى لها تعليمياً ، فإن الممارسة الفعلية تستدعى ممارسة وفهماً أعمق وأشمل مما تتطلبه هذه التمارين . وذلك مرجعه إلى طبيعة الملابس والظروف التى تصاحب تطوير البرامج لحل مشاكل واقعية .

وسيحاول الكتاب عن طريق الحالات العملية إعطاء المتعلم صورة هى أقرب ماتكون إلى الواقع . وبيت القصيد هنا أن لا يكون القارئ انطباعاً معيناً بعد الاطلاع على الأمثلة والتمارين ، خاصة فى الفصول الأولى ، بعدم جدوى اتباع هذا الأسلوب وبالتالى التقليل من أهميته . وستتم مناقشة بعض النقاط التى لها علاقة بهذا فيما بعد .

طبيعة دور الوظائف والأجزاء الخاصة بها :

من الوظائف ماتكون أساسية ومنها ماتكون فرعية . وتحقيق الهدف من الوظيفة الأساسية يسبقه تحقيق الأهداف المتعددة للوظائف الفرعية ويخصص لكل وظيفة جزء محدد .

وطبيعة الوظيفة تحتم الدور الذى يلعبه الجزء الخاص بتلك الوظيفة، فالوظيفة الأساسية لها جزء التحكم Control Module والوظيفة الفرعية لها جزء التنفيذ Execution Module وتتفاوت مستويات التفرعات للوظائف حسب المتطلبات . ويوضح الرسم التجريدى التالى هذه النقطة .



وقد يكون من الضرورى، وإن كان من غير المستحب، أن يقوم الجزء بوظيفة تنفيذية ووظيفة تحكم . فقد يقوم الجزء بمعالجة تتطلب تفرعاً إلى جزء فرعى آخر أو أكثر، ومن ثم الاستمرار فى المعالجة . ومن الأفضل على كل حال أن يتلافى هذا الوضع عن طريق إعادة تركيب التفرعات، وإضافة مستوى آخر من هذه التفرعات .

حتى تتوقف التجزئة :

إن التحكم في مستويات الأجزاء والوظائف التابعة لها أمر يخضع لاعتبارات شخصية ، وليس لقوانين محددة تنطبق تلقائياً وفي كل الحالات .

وبالرغم من ذلك فإن هناك إرشادات عامة يمكن اتباعها للوصول بالتجزئة الوظيفية إلى مستوياتها المناسبة ، وذلك بالإجابة على السؤالين التاليين :

— هل يقوم الجزء بوظيفة محددة واضحة بحيث إن الاستمرار بتجزئتها قد يؤدي إلى تشويه طبيعة هذه الوظيفة ؟

— هل من الممكن برمجة الوظيفة التي يقوم بها هذا الجزء بسهولة ودون أن يتجاوز عدد عبارات البرمجة حداً أعلى (٥٠ عبارة مثلاً) ؟ ويحدد هذا العدد مسؤولية مركز الحاسب الآلى ليتبعه من قبل جميع ذوى الاختصاص في تطوير البرامج .

ومن الأخطاء الشائعة التي يقع فيها بعض المتعلمين عند التفكير في طبيعة وظائف الأجزاء : تقسيمها إلى أجزاء تقوم بعملية إدخال البيانات ، وأجزاء تقوم بعملية المعالجات ، وأخرى تقوم بعملية إخراج المعلومات . وقد مر ذكر ذلك في الفصل الأول ، حيث إن هذه هي الدورة الرئيسية التي تتبعها خطوات معالجة البيانات على الحاسب . ولكن ذلك لا يعنى بالضرورة أن تكون الأجزاء قد حددت بناء على هذه الوظائف الأساسية .

فالجزء قد يقوم بوحدة أو أكثر من هذه الوظائف الثلاث : الإدخال ، والمعالجة ، والإخراج ، وذلك حسب ما تقتضيه طبيعة الوظيفة الكلية لأي برنامج .

ويكون التركيز هنا على طبيعة المعالجة دون عمليات الإدخال والإخراج . وكما مر ذكره ، فإن المعالجة تتكون في أساسها من العمليات الحسابية والمنطقية . وتتفاوت هذه في صعوبتها وفق طبيعة المعالجة : من العمليات الحسابية البسيطة إلى العمليات

الإحصائية التنبئية المعقدة . وقد تجمع كل عمليات الإدخال والإخراج في جزأين منفصلين ، وذلك إذا تطلبت المركزية في تنفيذها . وفي الأحوال الأخرى ، وبشكل عام ، فإن توزيع عمليات الإدخال والإخراج إلى الأماكن التي تتم فيها المعالجات التابعة لها ، يؤدي إلى الوضوح في طبيعة الوظيفة .

طبيعة التفاعل فيما بين الأجزاء :

يتم التفاعل والاتصال بين الجزء التنفيذي والجزء المتحكم بواسطة نوعين رئيسيين من المعلومات : المؤشرات والبيانات .

المؤشرات : تأخذ المؤشرات إحدى قيم متعددة ، وترمز كل قيمة منها إلى طبيعة الوظيفة الفرعية والجزء الخاص بها والذي سيتم التفرع إليه . وقد تستخدم المؤشرات كأسلوب للتحكم في إصدار أمر بدء التنفيذ من الجزء المتحكم ، وإشعار الانتهاء من التنفيذ من الجزء التنفيذي . وفي الحالة الأخيرة ، تكون القيم ثنائية ، إما «نعم» أو «لا» ، فتكون القيمة «لا» عند إصدار أمر التنفيذ ، وتتحول إلى «نعم» عند الانتهاء منه ، وقد تمثل القيمتان بأرقام ، «.» و «١» مثلاً .

البيانات : تتراوح كمية البيانات المتبادلة بين الأجزاء من الحقل (الوحدة الأساسية) إلى السجل . ومن الممارسات غير السليمة أن يتم استعمال ملف معين من قبل أجزاء متعددة ودون تخصيصها ، بحيث يصعب ملاحقة ومتابعة التغيرات التي تطرأ على الملف .

وكذلك فإن انسياب المعلومات بين الأجزاء يكون في اتجاهين ، من الجزء المتحكم إلى الجزء المنفذ وبالعكس . فالبيانات المرسلة من الجزء المتحكم إلى الجزء المنفذ ، تحول إلى بيانات مغرجة ، ومن ثم يعاد إرسالها إلى الجزء المتحكم . وإذا كان هناك ضرورة في جعل هذه البيانات مشتركة بين أجزاء متعددة في النظام ، فليس ثمة حاجة إلى نقلها من جزء إلى جزء ؛ إذ قد تجعل مشاعة في النظام ككل .

وإذا كانت المعلومات المتراصة بين جزأين هي المؤشرات ، فإن الجزء المتحكم يجعل قيمة المؤشر بإحدى قيمتيه «لا» مثلاً — وبعد انتهاء الجزء المنفذ من جميع المعالجات الخاصة به ، يقوم بتغيير قيمة المؤشر إلى القيمة الأخرى له — «نعم» مثلاً — معلناً بذلك انتهاء التنفيذ والعودة إلى الجزء المتحكم .

وبشكل عام يفضل التقليل من كمية البيانات المشتركة والمتحركة بين الأجزاء . ولكن قد يكون من الصعوبة بمكان ، كتابة البرامج الجزئية مستقلة تماماً إلى درجة انعدام أى تفاعل فيما بينها .

وفى كل الأحوال لابد من تحديد التفاعل والتشارك بين الأجزاء وتوثيقها بطريقة ما ، سواء فى البرنامج نفسه أو فى التوثيق الملحق به .

حجم الجزء :

كان حجم الجزء يقاس تقليدياً بمقدار حجم الذاكرة بالحروف والذى يمكن أن يحتله الجزء فى الذاكرة . أما حديثاً ، ومع انخفاض تكلفة الذاكرة ، فإن حجم الجزء يقاس بعدد العبارات البرمجية القابلة للتنفيذ ، أى باستثناء الملاحظات والتوجيهات والإيضاحات . وهناك مقياس آخر أدق يستند إلى هذا الأخير ، وهو عدد الأسطر فى البرنامج بعد انضغاطه بإزالة بعض الحروف والفراغات .

وقد أشير إلى أن الاستمرار فى التجزئة يتوقف إذا كانت الوظيفة قابلة للبرمجة فى حدود (٥٠) سطراً من عبارات البرمجة . وينطبق هذا على الأجزاء الفرعية التفصيلية التى تظهر فى البرنامج . لكن قد يكون حجم الجزء أكبر بكثير من ذلك ، استناداً إلى مدى شمولية النظام .

ففى الشركات الكبيرة تقاس الأجزاء بآلاف السطور من عبارات البرمجة المنضغطة .

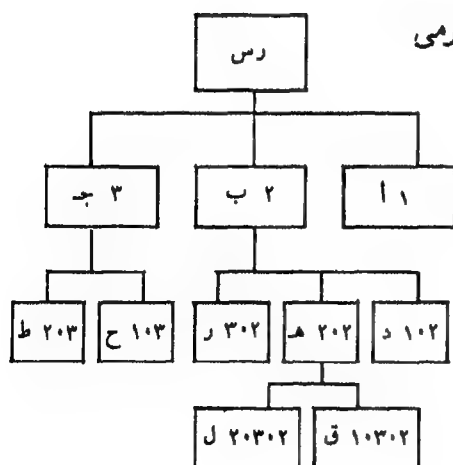
والتفريق هنا هو بين الجزء التابع لبرنامج والجزء التابع لوظيفة رئيسية في شركة كبيرة . فالجزء الخاص بحساب مجمل الراتب هو جزء برمجى ، والجزء الخاص بإنتاج حسابات شؤون الموظفين هو جزء رئيسى نتج عن ضرورة تحديد العلاقات بين الأجزاء الرئيسية الكبيرة في مؤسسة ما . وهذا الجزء الأخير يكون في العادة من نتاج التحليل والتصميم ، أما الجزء البرمجى فقد ينتج عن نفس المرحلة أو مرحلة البرمجة . ذلك أن المبرمج قد يرى ضرورة وجود أجزاء صغيرة تقوم بوظائف بسيطة لتسهيل عملية كتابة وتنفيذ البرامج .

الأساس الثانى : التركيب الهرمى (TOP-DOWN) :

إن الوظائف الرئيسية والفرعية والأجزاء الخاصة بها لابد أن تتبع نظاماً محدداً لتنظيم العلاقات والتفاعل فيما بينها . ويتم تحقيق ذلك بواسطة التركيب الهرمى الذى يشبه الشجرة المقلوبة ، كما مر ذكره في الفصل السابق . وستتم الآن مناقشة بعض النقاط التى لها علاقة بهذا الأساس .

طرق رسم الهيكل الهرمى :

يبين الشكل التالى رسماً تجريبياً لتركيب هرمى



شكل (١-٣)

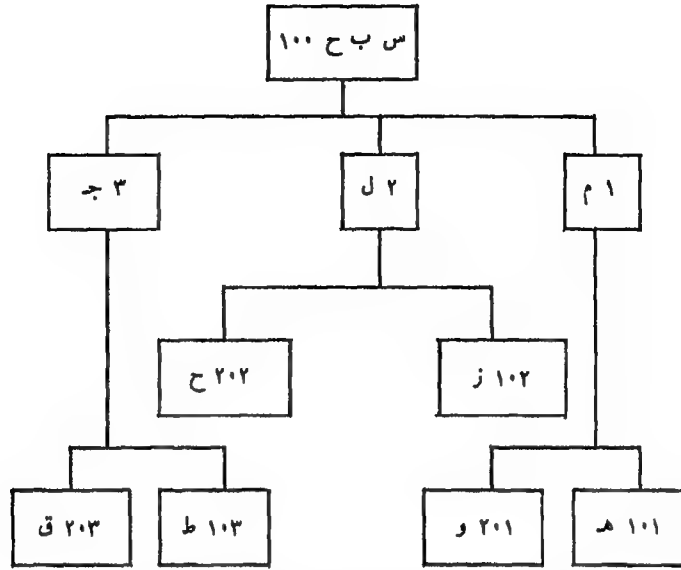
مثال تجريدى لهيكل هرمى .

لاحظ النظام الذى اتبع فى ترقيم الأجزاء لتسهيل الإشارة إليها وربطها بعضها مع بعض . كذلك قد يعطى كل جزء اسماً مختصراً أو رمزاً معيناً مكوناً من أحرف و/ أو أرقام ، حيث يتم استعمال هذا الاسم والرمز فيما بعد عند كتابة * البرامج . والأحرف المستخدمة فى أشكال التركيب الهرمى هى رمزية ، ويمكن أن تكون فى أى تسلسل ، ومكونة فى أغلب الأحيان من أحرف متعددة تشير إلى طبيعة الوظيفة .

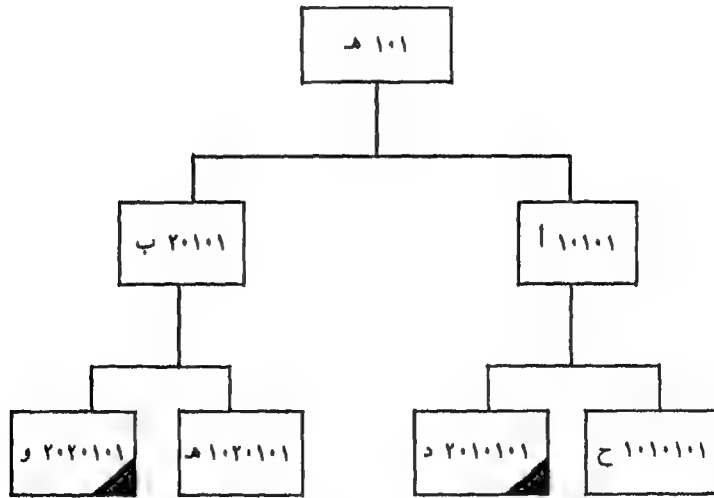
أما فيما له علاقة بعدد المستويات ، فإن التعداد الزائد لها فى صفحة واحدة قد يؤدى إلى تعقيدها وبالتالى صعوبة فهمها ومتابعتها ، ولا يوجد عدد محدد متفق عليه للمستويات فى الصفحة الواحدة . وكإرشاد عام يمكن أن يكون الرقم من ٣ — ٧ مستويات عدا المستوى الرئيسى الأول الذى يحوى مربعاً واحداً . ويمكن أن يؤخذ بعين الاعتبار الشعب الأفقى ، إضافة إلى الشعب العمودى . وبشكل عام ، فإن الزيادة فى الشعب الأفقى تؤدى إلى التقليل من عدد المستويات (الشعب العمودى) للصفحة الواحدة . وتعطى الصفحات التالية أمثلة تجريدية بهياكل هرمية . لاحظ التسلسل فى أرقام الأجزاء فى كل صفحة ، والعلاقة فيما بينها على الصفحات المتعددة .

ومن فوائد هذا التقسيم أن ذوى العلاقة يطلعون على المستوى التفصيلى للتركيب الهرمى الملائم لاحتياجاتهم . فالمستفيد والمحلل والمصمم قد ينصب اهتمامهم على المستويات الشمولية ، فى حين ينصب اهتمام المبرمج على المستوى التفصيلى للتركيب الهرمى ، وخاصة عند صيانة البرنامج فى مرحلة التشغيل .

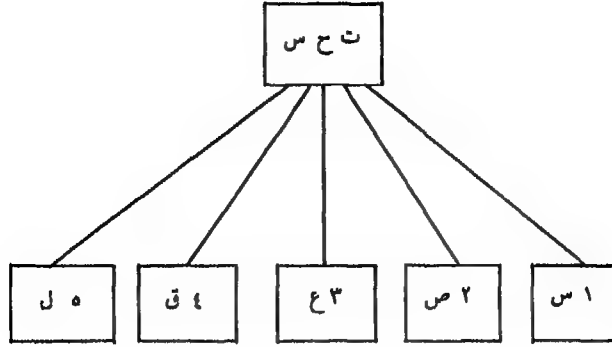
• عند كتابة البرامج بلغات البرمجة التى تستخدم عبارات باللغة الإنجليزية لمعالجة معلومات بالهرمية ، فإن هذه الأسماء والرموز تكون باللغة الإنجليزية .



شكل (٣ - ٢) المستوى الشمول الأول في التركيب الهرمي .

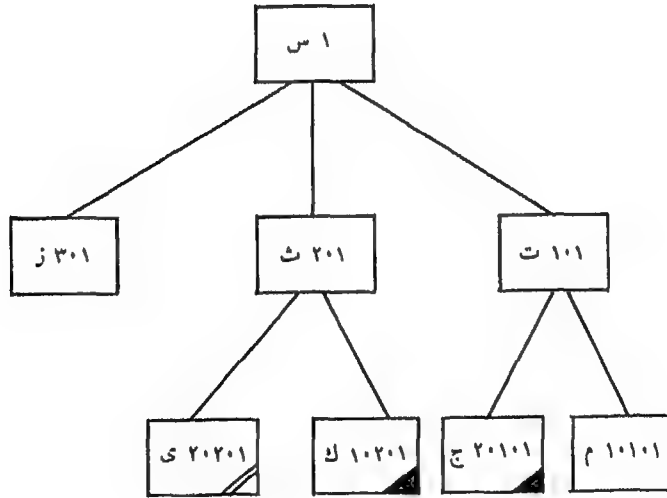


شكل (٣ - ٣) التركيب التفصيل الهرمي للجزء رقم ١٠١ هـ من الشكل (٣ - ٢) .
و يستمر في تفصيل بقية الأجزاء من الشكل (٣ - ٢) حسبما تقتضيه طبيعة المشكلة .



شكل (٣-٤) تركيب هرمي ذو تفرعات أفقية متعددة

لاحظ أنه إذا دعت الحاجة إلى مستوى آخر لواحد أو اثنين من الأجزاء يمكن دمجها هنا. كذلك لاحظ الأسلوب الآخر لوصل الأجزاء.



شكل (٣-٥) تفصيل الجزء ١ س التابع لشكل (٣-٤)

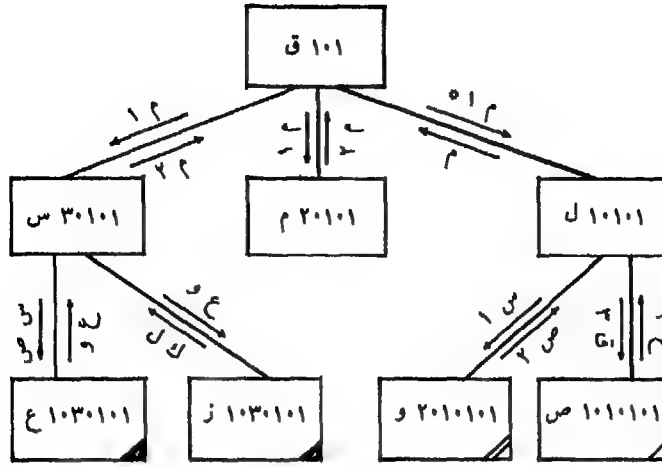
وهكذا لجميع الأجزاء الأخرى المبينة في الشكل (٣-٤) إذا اقتضى الأمر.

ولتسهيل المتابعة فلا بأس من إضافة شروحات مبسطة لطبيعة الوظائف التى يقوم بها الجزء والمثلة بالرموز الموضوعة داخل المربعات وذلك فى أسفل كل صفحة، و/ أو قد توضع فى قاموس موحد خاص بالنظام ككل . (Data Dictionary) .

الأجزاء التى يتكرر تنفيذها :

وإذا تكرر تنفيذ جزء من قبل أكثر من جزء متحكم، فهناك أساليب مقترحة لتبيان هذا التكرار : (أ) توضع إشارة موحدة للأجزاء المتكررة ذات الوظائف المختلفة فى إحدى الزاويتين السفليين، كما فى الشكل (٣ - ٣)، أى أن الجزء ٢٠١٠١١، والجزء ٢٠٢٠١١ ينفذان من قبل أجزاء متحكم، وليس من قبل جزء واحد فقط بغض النظر عن طبيعة الوظيفة للجزء المتكرر، (ب) تستخدم إما خطوط متعددة أو أحجام مختلفة من التظليل لتوحيد الإشارة إلى الأجزاء المتكررة ذات الوظيفة الواحدة، وفى نفس المجموعة من التركيبات الهرمية . انظر إلى الأمثلة فى الشكل (٣ - ٥) . أى أنه إذا تمت الإشارة إلى الوظيفة التى يقوم بها الجزء ٢٠٢٠١ فى تركيب هرمى آخر، فإن عدد الشركات يكون اثنتين، كذلك فإن حجم الجزء المظلل من المربع الممثل للجزء ٢٠١٠١ يماثل المربع الممثل لأى جزء آخر يؤدى نفس الوظيفة، و (ج) الأسلوب الأخير هو المزج بين التظليل والخطوط لنفس المجموعة من التركيبات الهرمية . انظر إلى الأمثلة فى الشكل (٣ - ٦) . وقد يستخدم المصمم أو المبرمج نظاماً أخرى شبيهة بهذه .

كاستخدام عدد نقاط مختلفة أو رموز مختلفة (* - ◊) داخل المثلث فى زاوية المربع .



شكل (٣-٦)

استخدام مزيج من الأساليب للإشارة إلى الأجزاء المتكررة
التي تقوم بنفس الوظيفة في تركيب هرمى تفصيلى .

تحتوى المربعات الأخرى المثلة للوظيفة ز نفس الحجم المظلل .

• هناك إشارات ورموز أخرى سيتم التحدث عنها تبعاً .

ويمكن الاكتفاء باستخدام نفس الرمز للوظيفة الواحدة المتكررة للحفاظ على وحدة
الإشارة إليها في التركيبات الهرمية المختلفة التابعة لنظام واحد .

البيانات المشتركة بين الأجزاء :

ومن الاتجاهات الحديثة في رسم التركيب الهرمى، تبيان البيانات على الخطوط
الموصلة بين الأجزاء . وهناك تعبير جديد يطلق على التركيب الهرمى الموضح عليه
انسياب البيانات وهو «الخريطة التركيبية» (Structure Chart) .

وكمثال على ذلك، انظر إلى الشكل (٣-٦) ولاحظ الإشارات والكتابة فوق
الخطوط الموصلة بين الأجزاء . وتدل الرموز المبينة فوق الخط م ١ على اسم البيانات

المنفولة من وإلى الجزء المتحكم . أما النقطة السوداء في نهاية الخط فتشير إلى اتجاه انسياب البيانات . ومن الأفضل تبيان البيانات التي تتم معالجتها فقط في الجزء المنفذ .

فمثلاً ، إذا كانت البيانات المنفولة سجلاً كاملاً للموظف ، ولكن الجزء المنفذ يستخدم حفل الدرجة والمرتبة فقط ، يجب ذكر الحقلين الأخيرين فقط .
و يؤدي اتباع هذا الأسلوب إلى الوضوح في الخريطة التركيبية وسهولة متابعة وظائف أجزائها .

التسلسل في تنفيذ الأجزاء :

تخضع الأجزاء حين تنفيذها إلى أحد أسلوبين : الاختياري ، والمتتالي . فأما الاختياري فيفضى بأن الجزء المتحكم يختار أحد الأجزاء المنفذة دون الآخر . وإذا كان الأمر كذلك ، يوضع رمز الاختيار في أسفل منتصف المربع الممثل للجزء . انظر إلى الجزء ١٠١ - ق في الشكل (٣ - ٦) وشرح ذلك ، أن الجزء «ق» ينفذ أحد الأجزاء الثلاثة «ل» أو «م» أو «س» بناء على قيمة المتغير م ١ ، وترسل قيمة محددة للبيانات م ٢ من الجزء المنفذ إلى المتحكم .

وأما في أسلوب المتتالي ، فيتم تنفيذ الأجزاء المتشعبة من جزء متحكم بالتتالي . ففي الشكل (٣ - ٦) ينفذ الجزء «و» ومن ثم الجزء «ز» . وفي بعض الحالات قد يكون المخرج من جزء منفذ هو بذاته المدخل إلى جزء آخر . ففي نفس الشكل ، نجد أن البيانات «ع» هي مخرجات من الجزء «ع» ومدخلات إلى الجزء «ز» .

بالإضافة إلى ذلك ، فإن الانتقال من جزء إلى آخر يجب أن يتم عبر السلم الهرمي للأجزاء . فلاننتقال من الجزء «ز» إلى «و» ، يجب اتباع المسار التالي :

ز ← س ← ق ← ل ← و .

خصائص التجزئة الوظيفية الهرمية :

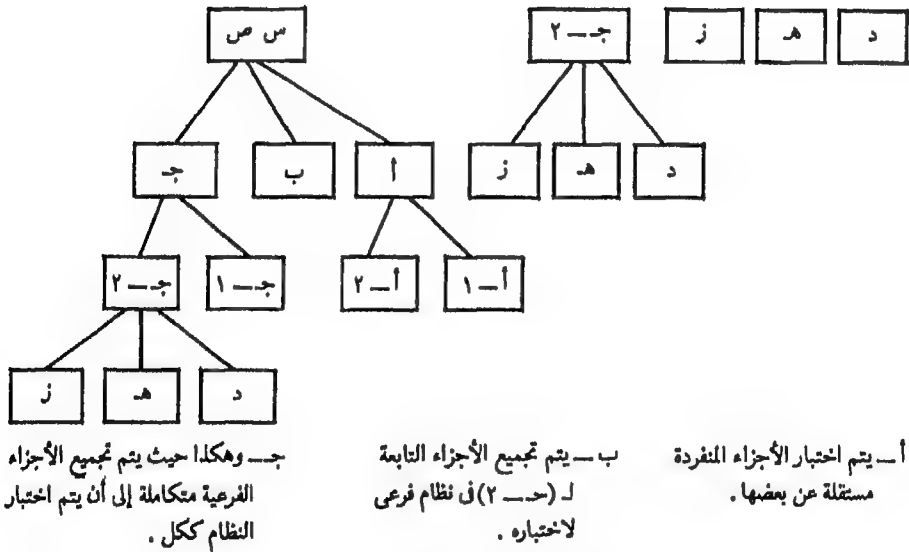
- كما سبق ، يتبين أن من خصائص التجزئة الوظيفية الهرمية ما يلي :
- ١ — تساعد على تفتيت المشكلة المعقدة الكبيرة إلى وحدات صغيرة يسهل التحكم فيها .
 - ٢ — تعطى صورة واضحة عن النظام قيد الدراسة ، سواء كانت شاملة أو تفصيلية ، وذلك حسب اهتمام المطلع عليها .
 - ٣ — تبين الوظائف الأساسية والفرعية وكيفية ترابطها فيما بينها .
 - ٤ — تعطى صورة عن الوضع الحالى للنظام ، وذلك لسهولة تحديثها وصيانتها ، سواء بالإضافة أو بالحذف أو بالتعديل .

فوائد التجزئة الوظيفية الهرمية :

- لأسلوب التجزئة الوظيفية الهرمية فوائد متعددة :
- ١ — تساعد في عملية تطوير وكتابة البرامج . وتتم هذه العملية بإحدى خطتين : التنفيذ من أعلى إلى أسفل ، أو من أسفل إلى أعلى .
- والتنفيذ من أعلى إلى أسفل يتطلب كتابة البرامج للأجزاء العلوية ، متفرعاً إلى المستويات التنفيذية السفلية ، وعند كل مستوى يتم استخدام برامج وهمية لأجزاء المستوى الذى تحته ، إلى أن يتم إنهاء جميع الأجزاء . ففى الشكل (٣-٤) مثلاً ، يتم تطوير البرنامج للأجزاء ت ح س ، س ، ص ، ع ، ق ، ل كمرحلة أولى . وبعد ذلك قد تتبع البرمجة أحد اتجاهين : إما أن يتم التطوير لجميع الأجزاء فى المستوى التالى ، مستوى ت ، ث ، ز فى الشكل (٣-٥) ، وهكذا التحرك أفقياً وإنهاء برمجة الأجزاء كلها فى مستوى معين . وإما أن يكون الاتجاه عمودياً ، كما فى الشكل (٣-٥) مثلاً ، حيث يتم الانتهاء من برمجة جميع أجزائه ، مستوى مستوى ، إلى آخر مستوى سفلى ، م ، ج ، ك ، ي ، فى هذه الحالة .

ومن جهة أخرى ، فإن التنفيذ من أسفل إلى أعلى يقضى بكتابة البرامج للأجزاء التنفيذية في أدنى المستويات ، حيث تجمع هذه الأجزاء بعضها مع بعض وفق التركيب الهرمى ، إلى أن يتم تجميع جميع الأجزاء للنظام ككل .

٢ — تسهيل عملية اختبار النظام . وكما في تطوير البرامج ، فإن اختبار النظام يتبع أحد الأسلوبين : من أعلى إلى أسفل ، أو من أسفل إلى أعلى ^١ .
وربما يكون من المستحسن اختبار نفس الأسلوب لتطوير واختبار النتائج . وإن كان ذلك ليس ضرورياً . ففى حالة اتباع أسلوب التطوير من أعلى إلى أسفل وبشكل عمودى ، يمكن البدء بالاختبار به من أسفل إلى أعلى للجزء من التركيب الهرمى الذى تمت كتابة البرامج له .



شكل (٣-٦) الاختيار من أسفل إلى أعلى

1. Edward Yourdan, «Techniques of Program Structure and Design» (Englewood Cliffs :
Printic Hall , Inc., 1975), p.58.

٣ — تسهيل عملية اكتشاف الأخطاء وتحديد آثارها . إن تقسيم المشروع الكبير المعقد إلى أجزاء وظيفية، وتقليل التفاعل فيما بينها، وتعريفه، يؤدي إلى سهولة اكتشاف موطن الأخطاء وتحديد مسار تأثير خطأ ما في جزء وظيفي على الأجزاء الأخرى . وكلما زاد مستوى التفاعل بين الأجزاء، كلما زادت معه صعوبة اكتشاف وعزل وتصحيح الأخطاء، وخاصة إذا مانتج هذا الخطأ عن تعديل في جزء ما في مرحلة التشغيل، للنظام ككل . ويطلق على انتشار تأثير الخطأ في جزء إلى الأجزاء الأخرى «العامل التموجي» .

بالإضافة إلى ذلك فإن تأثير التعديلات قد يسبب مشاكل أيضاً، فتعديل في طبيعة المدخلات أو طبيعة المعالجات قد يقتضي تعديلات أخرى في أجزاء لها علاقة بالجزء المعدل .

فبتحديد وظائف الأجزاء، وتقليل وتقنين التفاعلات فيما بين الأجزاء، يسهل عملية عزل واكتشاف الأخطاء، ويخفف من تأثير التعديلات في جزء على الأجزاء الأخرى .

٤ — تساعد في إنجاز عمليات الصيانة . يمكن تقدير أهمية هذه الفائدة إذا علم أن حوالى ٥٠% من الموارد المالية والبشرية المخصصة لتشغيل وتطوير البرامج تستهلك في صيانة البرامج الحالية^٢ . فالأجزاء الجديدة تضاف إلى الهيكل الهرمي في موقع مناسب، حيث تتوافق وتتجانس وظيفة الجزء الجديد مع وظائف الأجزاء الأخرى . أما طلبات الصيانة التي تستدعي تعديل جزء خالٍ، فيتم توجيهها إلى الجزء ذى العلاقة .

2 — B. P. Lebtz, E. B. Swanson and, G. E. Tompkins, "Characteristics of Application Software Maintenance," Communication of the ACM. Vol. 21, No. 6 (June 1978), 466-471.

وكذلك نتائج استبيان وزع على حوالى ٢٣ مركزاً رئيسياً للحاسبات في مدينة الرياض عام ١٤٠٥ هـ . انظر ندوة «التقنية الحديثة في تنظيم وإدارة المعلومات»، معهد الإدارة العامة، ٥ — ٦ ربيع الآخر ١٤٠٧ هـ، صفحة ٥٠ .

و يتم اختبار الجزء الجديد أو المعدل دون الحاجة إلى اختبار النظام ككل . وهذا كله يؤدي ، بالإضافة إلى التوفير في المال والوقت ، إلى التخفيف من العبء النفسى على مبرمجى الصيانة .

٥ — تستخدم كأداة لإدارة مشاريع التطوير، حيث يمكن استخدام الهيكل الهرمى كوسيلة للتحكم فى توزيع الصلاحيات ، ولتحسين وسائل الاتصال بين المجموعات المختلفة التى تقوم بتطوير نفس النظام ، ولتقويم نتائج المشروع ومدى تقدمه نحو تحقيق هدفه .

ففى المشاريع المعقدة الكبيرة، يمكن توزيع مسؤوليات تطوير الأجزاء على مجموعات مختلفة وفق التركيب الهرمى . ويتم الاتصال بين هذه المجموعات ؛ لتحديد أى تفاعل بين الأجزاء المطورة . وفى أية لحظة من لحظات تطوير المشروع ، يمكن لمدير المشروع أن يحصل على فكرة عامة عن مدى تقدم الأجزاء ، وعما إذا كان ماتم إنجازها من الأجزاء قد حقق الهدف من وظيفتها .

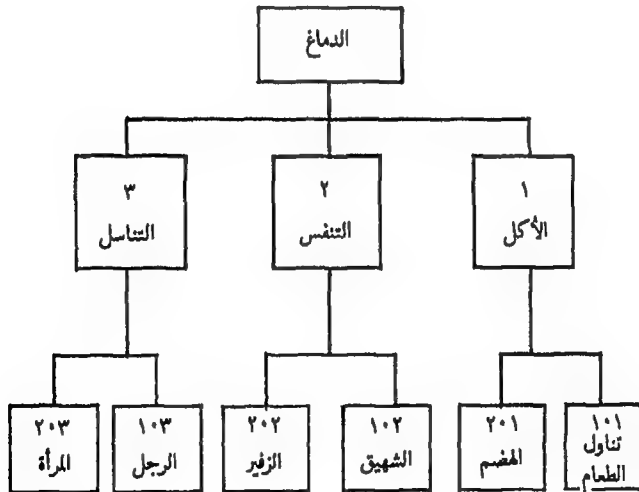
٦ — زيادة إنتاجية المبرمج بعامل مقداره ٣١,٦ .

أمثلة عملية

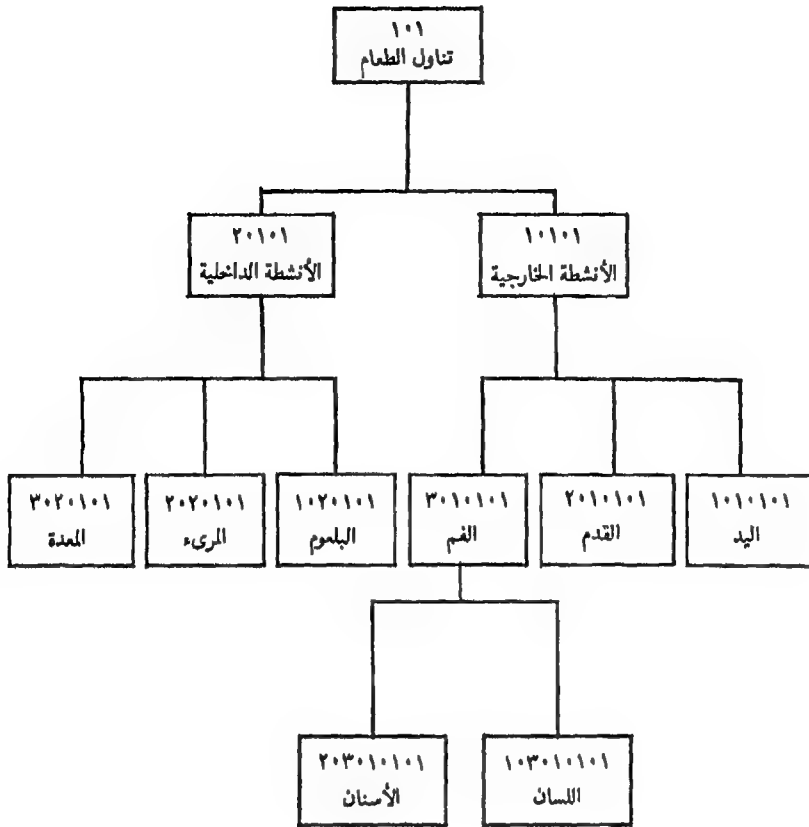
تقدم الصفحات التالية أشكالاً متعددة من التركيب الهرمي لأمثلة مختلفة .
وسيكون تسلسل الأمثلة كالتالى :
الإنسان ، مثال حكومى ، مثال تجارى .

الإنسان

يؤدى الإنسان وظائف متعددة ومتداخلة ، وهناك وظائف أساسية كالتنفس والأكل والتناسل ، وهناك وظائف فرعية تقوم بها أجزاء تابعة لكل وظيفة رئيسية كالأيدى والأنف واللسان . ويعطى الشكل (٣ - ٧) الهيكل الهرمى العام للإنسان ، ويمثل الشكل (٣ - ٨) الهيكل الهرمى التفصيلى لوظيفة الطعام .



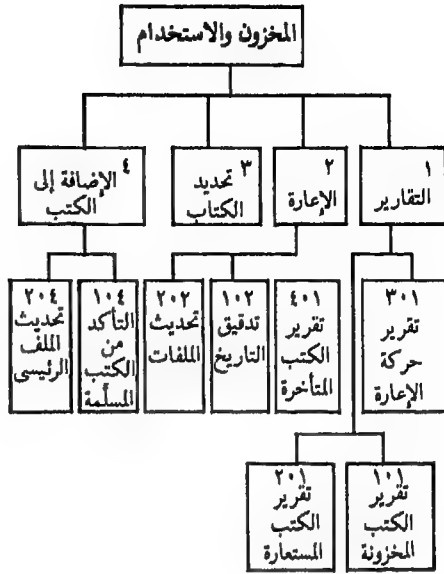
الشكل (٣ - ٧) الهيكل الهرمى العام لوظائف الإنسان



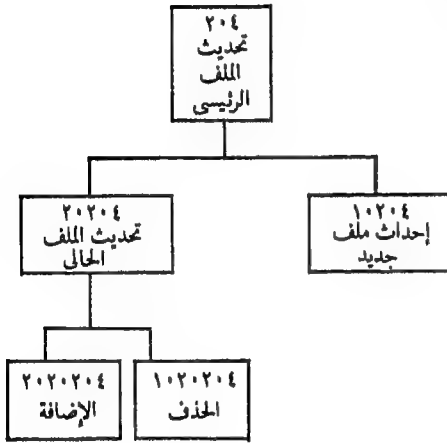
شكل (٣-٨) الهيكل الهرمي التفصيلي للجزء ١٠١ «تناول الطعام»

مثال حكومي :

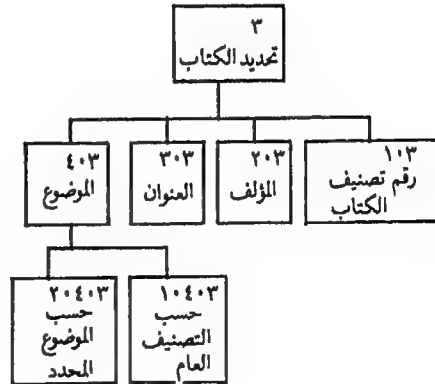
تحتوي مكتبة مدرسة حكومية ثانوية على ما يقارب ٢٠,٠٠٠ كتاب، وتتبع تصنيف ديوي . وقد قام فريق بتحليل عمليات وأنشطة المكتبة، وتم تصميم عدة هياكل هرمية متسلسلة لوصف الوظائف الرئيسية في النظام، كما هو مبين في الأشكال (٩-٣)، (١٠-٣)، (١١-٣) .



شكل (٣ - ٩)
الهيكل الهرمي لوظائف المكتبة .



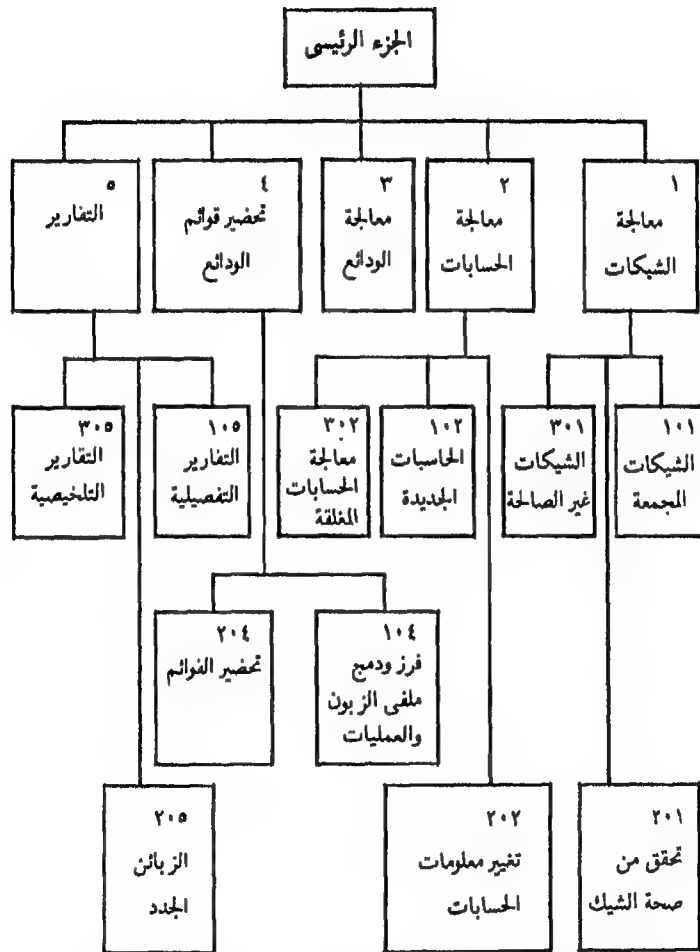
شكل (٣ - ١٠)
الهيكل الهرمي التفصيلي للجزء ٢٠٤
«تحديث الملف الرئيسي» .



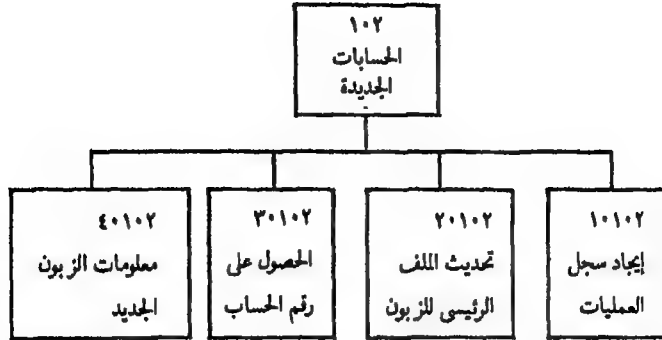
شكل (٣ - ١١)
الهيكل الهرمي التفصيلي
للجزء ٣، «تحديد الكتاب» .

مثال تجارى :

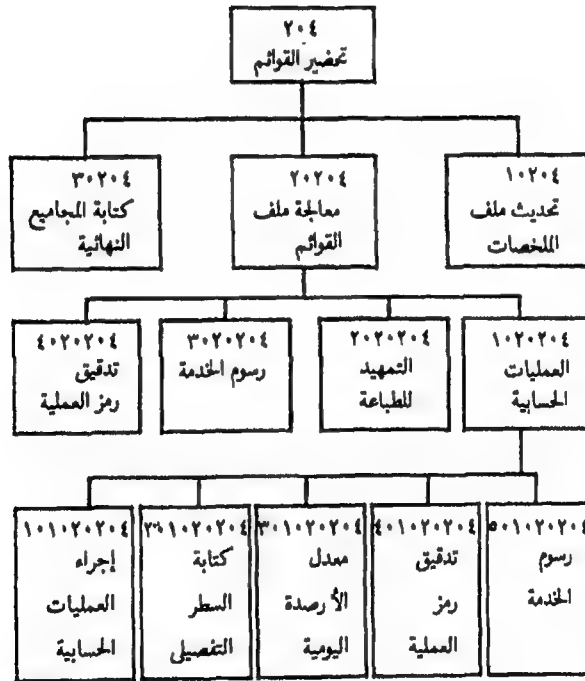
يقدم البنك س ص ع خدمات متعددة، وقد أرادت الإدارة إضافة خدمة جديدة
لنوع معين من الحسابات التى تحمل فائدة، وتخضع لقواعد محددة . وقد تم تحديد عدة
هياكل هرمية بوصف وظائف النظام وعلاقاتها بعضها مع بعض ، ويحتوى الأشكال (٣ -
١٢)، (٣ - ١٣)، (٣ - ١٤) بعضاً منها .



شكل (٣ - ١٢) الهيكل الهرمى العام لمعالجة الحسابات الجارية الخاصة ببنك س ص ع .



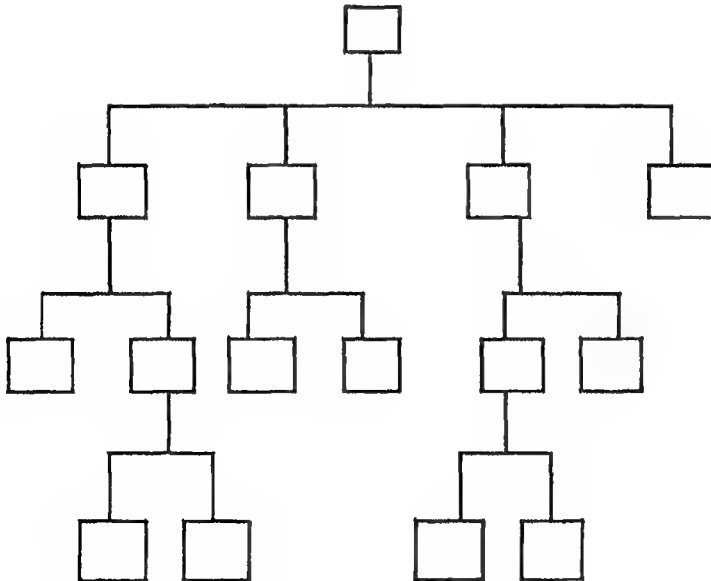
شكل (٣-١٣) الهيكل الهرمي التفصيلي للجزء ١٠٢ «الحسابات الجديدة» .



شكل (٣-١٤) الهيكل الهرمي التفصيلي للجزء ٢٠٤ «تخصير القوائم» .

تمارين

- ١ - ماهى طبيعة تسلسل تنفيذ الأجزاء ؟
- ٢ - اشرح باختصار أنواع الأجزاء فى الهيكل الهرمى .
- ٣ - اشرح باختصار الأساسين الرئيسيين للتجزئة الهرمية .
- ٤ - ماهى أساليب تطوير (كتابة) البرامج للأجزاء المبينة فى الهيكل الهرمى ؟
- ٥ - ماهى أساليب اختبار الأجزاء فى الهيكل الهرمى ؟
- ٦ - للهيكل الهرمى التالى أجب عن الأسئلة الآتية :
 - أ - رقم الأجزاء ، وأعط أسماء لها مكونة من حرفين .
 - ب - حدد الأجزاء التنفيذية والأجزاء المتحركة .
 - ج - كيف يتم تطوير البرامج لهذه الأجزاء ؟ اذكر جميع الأساليب المتوفرة .



٧ — التعليمات : ارسم الهيكل الهرمى للحالات التالية حسب ماتراه مناسباً ، سواء في مستوى واحد أو مستويات متعددة . ضع فرضيات مقبولة لأى معلومات ناقصة .
أ — تشتمل عمليات الطلبيات في محل (هارب للأسماك) على النشاطات الرئيسية التالية :

يقدم الزبون طلباً ، ومن ثم يرى إذا كان في الإمكان منحه الذئب المطلوب في الطلبية . أما عمليات البضائع فتتكون من فاتورة الزبون ، وأوامر الشحن للبضائع ، مذكرة رفض من الزبون ، وتحديث ملف المخزون . أما طلبية الزبون فتشمل رقم الحساب والأسماك المطلوبة ، ويتم إيجاد التكلفة وشروط البيع .

ب — يشتمل نشاط قسم خدمات الأغذية في مستشفى كبير على عمليتين رئيسيتين : تحضير الطعام ، والتحكم في عيارات الأكل عند المرضى ، والمرضى الذين لهم عادة (حمية) معينة إما أن يكونوا على برنامج عام للأكل أو خاص . والبرنامج الخاص يتطلب حساب متطلبات التغذية والكالوريات والبروتينات .

ج — تشتمل العمليات الرئيسية لمراقبة المخزونات في غازن تميم الكبيرة والمتعددة الفروع على التالى :

— التسعير .

— التسلم .

— الطلبات العامة .

— طلبات التنزيلات .

أما الطلبيات العامة فيمكن أن تكون في أحد الأشكال التالية :

— طلبيات الملابس عن طريق التذاكر الخاصة بها .

— طلبيات البضائع الأخرى من الموردين .

— الطلبيات الأتوماتيكية عن طريق جهاز الفحص الإلكتروني .

الفصل الرابع

مقدمة عن وسائل تصنيف البيانات والمعلومات فى الحاسب

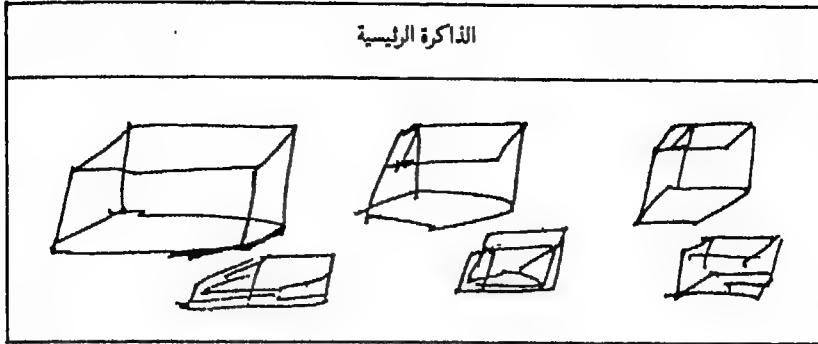
كما مر شرحه فى الفصل الأول ، فإن البيانات تدخل إلى الحاسب عن طريق أحد أجهزة الإدخال وتخزن فى الذاكرة ، ومن ثم تخضع للمعالجة الحسابية أو المنطقية أو كليتهما ، وتخرج النتيجة كمعلومات على إحدى وسائط الإخراج : شاشة ، طابعة ، أو طابعة مثلاً .

و يصحب عملية إدخال البيانات إدخال للبرنامج الذى يحدد الخطوات التى ستخضع لها البيانات بإحدى لغات البرمجة .

وتتبع طرق متباينة فى صعوبتها لوصف العمليات التى يقوم بها الحاسب عند القيام بالوظائف الأساسية الثلاث : الإدخال ، المعالجة ، والإخراج . وتزداد هذه الصعوبة كلما اقترب الوصف من عمل الحاسب الفعلى ، وتقل الصعوبة كلما اقترب هذا الوصف من أسلوب عمل المبرمج من خلال لغات البرمجة ، وتمشياً مع الأسلوب العام المبسط لهذا الكتاب ، فإن وصف هذه العمليات سيتم عن الطريق الأقل صعوبة وتعقيداً ، وهو من وجهة نظر المبرمج .

وصف عام لأسلوب تعامل الحاسب مع البيانات :

ينظر المبرمج إلى ذاكرة الحاسب وكأنها مقسمة إلى صناديق تخزن فيها البيانات . هذه الصناديق تحوى البيانات عند معالجتها الفعلية من قبل الحاسب ، أى عند تخزينها فى الذاكرة الرئيسية تمهيداً لمعالجتها . و يوضح الشكل التالى مفهوم الصناديق .



ومن الممكن القيام بالعمليات التالية للتحكم في محتويات هذه الصناديق :

١- إدخال وتخزين البيانات بداخلها، ويتم ذلك إما عن طريق وسائط الإدخال والأوامر التابعة لها، أو عن طريق التخزين المباشر في داخل البرنامج والأوامر التابعة له . والبيانات المدخلة عن طريق وسائط الإدخال إما أن تدخل عن طريق لوحة المفاتيح التابعة للطرفيات، أو عن طريق وحدات الذاكرة الفرعية المساعدة : الأشرطة، والأقراص المغنطة . ففي الحالة الأولى يكون الإدخال في وضع التخاطب مع الحاسب، وفي الحالة الثانية يتم تخزين البيانات على الأشرطة والأقراص المغنطة أولاً، ثم تسترجع بعد ذلك لمعالجتها من قبل البرنامج . ويتم التخزين الأولى هذا إما عن طريق أجهزة خاصة تخزن البيانات في الذاكرة الفرعية مباشرة من لوحة المفاتيح، أو قد يكتب برنامج خاص لذلك الغرض .

٢- القيام بالعمليات الحسابية : إذا احتوت هذه الصناديق على بيانات رقمية، يمكن إخضاعها للعمليات الحسابية المختلفة، من جمع وطرح وقسمة وضرب وأس .

ويمكن أن تؤلف محتويات الصناديق قيماً في معادلات معقدة وطويلة، حيث يقوم الحاسب بتقسيم هذه العمليات إلى سلسلة من الحسابات المرحلية وفق أسس معروفة ستناقش لاحقاً في هذا الفصل .

٣- نسخ وتبديل محتويات الصناديق من البيانات : من الممكن نقل محتويات صندوق إلى آخر، أو تغيير محتويات الصندوق الواحد، أو تبادل القيم بين صندوقين .

٤- إخراج محتويات الصناديق إلى الأجزاء الملحقه بالمعالج : سواء كان ذلك إلى الشاشة التابعة للطرفية، أو إلى آلة الطباعة، أو إلى وسائط التخزين المساعد .

كيفية التحكم في البيانات المخزنة في الصناديق :

يتسم هذا التحكم عن طريق إتاحة الفرصة للمبرمج بإعطاء أسماء لهذه الصناديق تتكون من أحرف وأرقام . وتختلف قواعد تسمية هذه الصناديق من لغة إلى أخرى، وإن كان الاتفاق بينها جميعاً أن تبدأ بحرف هجائي .

وعلى المبرمج أن يتابع محتويات هذه الصناديق وموافقتها لأسمائها إذ أن الحاسب لا يربط منطقياً بين الاسم والمحتوى . فاسم الصندوق CAT (أى قط) مثلاً لا يعنى بشئاً أن الحاسب لا يتوقع تخزين اسم قط بداخله . فصحة الموافقة بين المحتوى والاسم هى بالكلية من مهمة المبرمج، لذلك فإن من النصائح العامة فى مجال تسمية الصناديق أن يتوافق معنى اسم الصندوق مع محتواه، فمثلاً اسم «كمية» سيحتوى على كمية البضاعة المطلوبة، اسم «الرصيد» سيحتوى على الرصيد القابل للدفع من الزبون، ... وهكذا .

ومن الممكن أن يشير الاسم الواحد إلى صندوق واحد أو أكثر، وإذا أشار الاسم إلى أكثر من صندوق، فقد تكون هذه الصناديق مرتبة وفق قائمة ذات بعد واحد، مثل مجموعة الأسماء التابعة لمؤسسة معينة، فاسم المؤسسة هو اسم مجموعة الصناديق، وتحتوى الصناديق المنفردة — كل واحدة منها — على اسم لشخص معين . وقد يكون ترتيب الصناديق المتعددة لاسم واحد وفق جدول ذى بعدين، مثل المبيعات الأسبوعية من (الشاورما) لثلاثة فروع موزعة فى مدينة الرياض .

وتظهر هنا مشكلة في كيفية تخصيص وتحديد أحادي الصناديق التي تشترك في اسم واحد، وعولجت هذه المشكلة عن طريق إعطاء أرقام لأحادي الصناديق تتبع الاسم المشترك لها ومحاطة بقوسين، فإذا كان الاسم المشترك للصناديق ذات البعد الواحد هو «معهد» فإن الصندوق الأول يرمز إليه «معهد (١)»... وهكذا ويعكس الرقم التالي هذا الأسلوب :

اسم مجموعة الصناديق : «معهد»

.....						
	٦	٥	٤	٣	٢	١

رقم الصندوق

أما إذا كان ترتيب البيانات ذا بعدين، فيستخدم رقمان بينهما فاصلة «،» حيث يمثل كل رقم بعداً محدداً . وأقرب مثال لأذهان القراء لتمثيل هذه الحالة هي في تسمية أرقام الغرف في الفنادق ذات الطوابق المتعددة، حيث يتكون رقم الغرفة من جزأين الأول لرقم الطابق، والثاني لرقم الغرفة في ذلك الطابق . ونلاحظ اشتراك الغرف في الطابق الواحد في رقم الطابق أفقياً، واشتراك الغرفة ذات الموقع الواحد في الطوابق المختلفة في رقم الغرفة عمودياً . والاختلاف بسيط بين هذا الأسلوب في تسمية الغرف، وبين الأسلوب الذي يجب اتباعه مع الحاسب من قبل المبرمج . ففي الحاسب يجب البدء من الطابق العلوى إلى الطابق السفلى، أى بعكس الاتجاه في مثل الفندق، ويعكس الرسم التالى (صفحة ٩٧) المثل السابق حول المبيعات.

كما يلاحظ أن رقمى أحادي الصناديق، الأفقى والعمودى، تغزل بينهما الفاصلة «،»، وبناء على ذلك فإن شاورما الرياض (٢، ٢) يعكس مبيعات الأسبوع الثانى لفرع العليا، أى ١٠١ .

اسم الصندوق «شاووما الرياض»

الفرع المسجل	البطحاء	العليا	أم الحمام
١	١٤١	٢٤١	٣٤١
	٥٥	٦٢	٣٥
٢	١٤٢	٢٤٢	٣٤٢
	٦٨	١٠١	٥٣
٣	١٤٣	٢٤٣	٣٤٣
	٧٢	٨٠	٣١
٤	١٤٤	٢٤٤	٣٤٤
	٧١	٧٥	٨١

وأخيراً يجب التفريق بين اسم الصندوق ومحتواه، فالاسم ثابت لا يتغير، في حين أن محتواه — رقم المبيعات (١٠١) في هذا المثال — يتغير من شهر لآخر.

أصناف البيانات :

تختلف طبيعة البيانات المخزنة في هذه الصناديق باختلاف الهدف من حفظها وطريقة الاستعمال .

الهدف من الحفظ : إن الهدف من الحفظ قد يكون حسابياً أو غير حسابي، فالبيانات الحسابية تستخدم في المعادلات الجبرية والمقارنات، أما غير الحسابية فيتعامل معها الحاسب كحروف متتابعة من الممكن أن تطبع كما هي أو تستخدم في المقارنة فيما بينها، كفرز قائمة الأسماء حسب ترتيب الحروف الهجائية .

فالصندوق الحسابي يجب أن يحتوي على أرقام فقط دون حروف الهجاء والحروف الخاصة، أما الصندوق غير الحسابي فيمكن أن يحتوي على خليط من نوع أو أكثر من الحروف الرقمية والهجائية والخاصة .

وكثيراً ما تحتوى قواعد لغة البرمجة فى إطلاق الأسماء على الصناديق — على قاعدة محددة تميز محتويات البيانات كإضافة إشارة «\$» فى نهاية اسم الصندوق للدلالة على أن محتوى الصندوق هو غير حسابى .

طريقة الاستعمال : إن البيانات المخزنة فى الصناديق إما أن تكون ثابتة أو متغيرة، فالثابت من البيانات لا تتغير قيمته عند تشغيل البرنامج من فترة لأخرى . وهذا الثابت قد يكون حسابياً أو غير حسابى كذلك، فالحسابى منه قد يحزن فى صندوق ذى اسم معين، أو يثبت فى البرنامج كرقم محدد . ومن الأمثلة البسيطة على الثابت الحسابى هو الرقم «٢» الذى يستخدم لإيجاد مساحة المثلث بعد ضرب القاعدة فى الارتفاع . فأمنا اتباع لإحدى الحالتين التاليتين :

$$\frac{\text{القاعدة} \times \text{الارتفاع}}{2} = \text{المساحة} \quad \text{الحالة الأولى :}$$

$$2 = \text{القسم} \quad \text{الحالة الثانية : ١ .}$$

$$\frac{\text{القاعدة} \times \text{الارتفاع}}{\text{القسم}} = 2 \text{ . المساحة}$$

فيكون «القسم» هو الرقم (٢) والذى لابد من تخزينه ، بالإضافة إلى قيمته فى القاعدة والارتفاع، وذلك قبل التمكن من إيجاد المساحة . وكذلك البيانات المتغيرة قد تكون حسابية أو غير حسابية . فالحسابية منها تتغير من مرحلة تشغيلية إلى أخرى . ففى المثال السابق يكون الصندوقان «القاعدة» و «الارتفاع» محتويين على بيانات متغيرة، لذلك يطلق عليها الأسماء المتغيرة (Variable names) فى اصطلاح لغات البرمجة .

فعند تشغيل البرنامج للمرة الأولى لحساب مساحة مثلث ذى أبعاد محددة، تحزن هذه الأبعاد فى الصندوقين المذكورين، وإذا أراد شخص آخر إيجاد مساحة مثلث آخر ذى أبعاد مختلفة، يتم استخدام نفس الصندوقين لتخزين الأبعاد الجديدة ... وهكذا .

أما الثابت والمتغير من البيانات غير الحسابية فطريقة استعمالهما تشبه تلك الطريقة الخاصة بالحسابية منها ، فالبيانات المتغيرة غير الحسابية تتبدل قيمتها من عملية تشغيل لأخرى ، فأسماء الموظفين تخزن كل شهر مرة في الصناديق ذات الأسماء المتغيرة غير الحسابية ، فقد يكون هناك موظفون جدد قد أضيفوا إلى القائمة ، أو موظفون قد أحيوا إلى التقاعد قد حذفوا من القائمة ... وهكذا .

أما البيانات الثابتة غير الحسابية (Literals) فتعامل من قبل الحاسب وفق نصها الحرفي دون تحديد لطبيعة البيانات المخزنة ، فاسم الشركة الذى سيطبع على التقرير الشهرى لمبيعات «شاورما الرياض» ثابت لا يتغير مع التشغيل المتعدد على فترات زمنية ، وكذلك عنوانها إذا لزم الأمر .

وحكم الثابت غير الحسابى مثل الثابت الحسابى ، فقد تخزن البيانات غير الحسابية فى صندوق ذى اسم محدد ، فاسم الصندوق «شركة» قد يحتوى على «شاورما الرياض» ، وقد نطبع كلمتى «شاورما الرياض» مباشرة من البرنامج عن طريق وضعهما بين الإشارتين (« ») ، كما هو مكتوب فى الأسطر السابقة .

ويمثل الجدول التالى تلخيصاً لأصناف البيانات حسب الهدف من الحفظ وطريقة الاستعمال .

أصناف البيانات حسب طبيعة الاستعمال مع أمثلة

طريقة الاستعمال (طبيعة البيانات)	الهدف من الحفظ
متغيرة — تخزن فى صناديق لها أسماء الطول والعرض والكمية .	(طبيعة المعالجة) — حسابية .
ثابتة تظهر فى البرنامج ٢ ، ٣ ، ط	— غير حسابية .
«شاورما الرياض» «شركة داود للخزف»	الاسم ووصف المخزون والعنوان

أصناف العبارات التوضيحية :

إن مجرد إظهار البيانات ونتائج المعالجات على وسائط الإخراج المرئية من قبل المستخدمين مثل الشاشات والورق، لن يؤدي الغرض المنشود منها . فهذه البيانات غالباً ما تتكون من أرقام تمثل أشياء متباينة بحاجة إلى تعريف وتحديد . وقد تعرضنا في الجزء السابق لأحد هذه الأنواع من التعريف وهو الثابت غير الحسابي .

بالإضافة إلى ذلك، فإن تشغيل البرنامج قد يتطلب تعليمات توضيحية لما يجب أن يفعله المستخدم لتشغيل البرنامج بنجاح، ودون توقف أو انقطاع .

ونلاحظ أن هذه العبارات التوضيحية ثابتة لا تتغير مع التشغيل المتكرر للبرنامج، ومن أنواعها :

أ — العناوين (Labels) : يوضح العنوان مجموعة من البيانات التي ستدخل من قبل المستخدم، أو ستطبع من قبل البرنامج، فعبارة «المعلومات الشخصية عن الموظف» التي تظهر في أعلى الشاشة تفيد بأن البيانات التي ستدخل/ ستطبع لها صفة مشتركة .

وغالباً ما يستعمل العنوان لتعريف مجموعة من البيانات لها خاصية واحدة، ومرتبة إما أفقياً أو عمودياً . ويعطى الشكل التالي مثلاً لاستخدامات العناوين :

المبيعات الأسبوعية لفروع

شركة شاووما الرياض لشهر ربيع الأول ١٤٠٧ هـ

الفروع			
الأسبوع	البطحاء	العليا	أم الحمام
الأول			
الثاني			
الثالث			
الرابع			

فكلمة «الأول» تصنف مبيعات الأسبوع الأول للأفرع الثلاثة، في حين أن كلمة «البطحاء» تصنف مبيعات فرع البطحاء للأسابيع الأربعة في شهر ربيع الأول لعام ١٤٠٧ هـ ... وهكذا .

ب - تعريف البيانات (Data Definitions) : في بعض الحالات يستدعى تعريف أحادى البيانات بدلاً من مجموعة من البيانات . فكلمة «الاسم» تدل على أن اسماً سيظهر عقبها أو تحتها ، ويعطى المثال التالى توضيحاً لهذا الاستخدام :

المعلومات الشخصية عن الموظف

الاسم :	العمر :
تاريخ الميلاد :	تاريخ التعيين :
عدد الأولاد :	

وقد يستخدم كلا المثالين السابقين لإدخال البيانات أيضاً .

ج - المذكرات (Comments) : تستخدم المذكرات بشكل رئيسى لتيسير وتسهيل عملية التخاطب بين المستفيد والبرنامج ، وغالباً ما يكون المستفيد قليل أو معدوم الخبرة في مجال الحاسب . وقد يؤدي أية إعاقة أو توقف للبرنامج أثناء التشغيل إلى ففدائه الثفة في قدرات الحاسب . وتأخذ المذكرات أنواعاً متعددة :

فمذكرة الأمر تعطى توجيهات لمستخدم النظام ، كمدخل البيانات ، وذلك حتى تتم عملية إدخال البيانات بالطريقة السليمة خالية من الأخطاء ، فعبارة :

«أدخل الاسم مكوناً من ١٥ حرفاً على الأكثر» تدخل ضمن هذا الصنف .

وكذلك تظهر عبارات الأمر في شاشات الاختيارات والتى تحدد مسار التنفيذ في البرنامج كالشاشة التالية :

شاشة الاختيارات الرئيسية

١ - إدخال بيانات الموظف الشخصية .

٢ - اطبع بيانات الموظف الشخصية .

٣ - الخروج من النظام .

أدخل الاختيار المطلوب :

تحدد ثلاثة اختيارات على المستخدم أن يطبع أحدها، ومن ثم يتم التشعب في البرنامج لتحقيق المطلوب .

ومذكرة الأخطاء تستخدم لتنبيه مستخدم البرنامج إلى أن خطأ معيناً قد وقع أثناء تشغيل البرنامج، فالبارة :

«الاسم يشتمل على أرقام، أدخل الاسم مرة أخرى خالية من الأرقام من فضلك»
فيها تنبيه للخطأ وتوضيح لما يجب فعله لتصحيحه . وقد تظهر هذه المذكرات مباشرة على الشاشة إذا كان التشغيل للبرنامج في الوضع التبادلي . وفي بعض الأحيان قد يتم تحضير قائمة بالأخطاء وطبعها جميعاً مرة واحدة، ومثال ذلك، تحضير قائمة بالقطع التي تكون أرصدها سالبة القيمة، أو قائمة بأسماء الموظفين الذين لا تتوافق تواريخ تعيينهم مع تواريخ ميلادهم .

والملاحظات لها وظيفة إعلامية حيث توضح للمستخدم حالات معينة يكون المستخدم بدونها حائراً في تفسير ما يجري، فالبارة :

«أرجو الانتظار قليلاً لحين الانتهاء من تلبية طلبك»

تزيح عن ذهن المستخدم الارتباك أثناء الانتظار لفترة وجيزة يحار معها فيما يعمل، وكذلك البارة :

«هل تريد تعديل السجل كما طلبت أدخل نعم أو لا»

تحدد مرحلة في عملية التحديث يطمئن معها المستخدم إلى الحاسب .

كذلك قد تظهر الملاحظات في أسفل وثيقة ما كتميز عميل أوزبون عن آخر .
فعبارة : «شكراً على مواظبتكم على دفع أرصدة حسابكم في وقتها»
لها مفعول إيجابى فى نفس الزبون وقد تشجعه على الاستمرار فى عادته تلك .

المعالجات الأساسية التى يقوم بها الحاسب :

كما مر ذكره، فإن المعالجات التى يقوم بها الحاسب تتكون من جميع العمليات الحسابية والمنطقية، ولا بد من الإشارة هنا إلى أن الحاسب يقوم بعملية حسابية واحدة أو عملية منطقية واحدة فى نفس الوقت، إذ لا يستطيع الحاسب أن يجمع بين عمليتين منطقيتين أو بين عملية حسابية وأخرى منطقية . وسنتعرض الآن لهذه المعالجات :

أ — العمليات الحسابية وأولويتها : بالإضافة إلى العمليات الحسابية الأساسية من جمع وضرب وقسمة وطرح، فإن الحاسب يستطيع تمييز عمليات الأسس والجذور، ومن الممكن استخدام الأقواس لتحديد تسلسل المعالجات الحسابية فى معادلة جبرية طويلة .

ومن الأمثلة على المعادلات الجبرية ذات المتغيرات المتعددة والتى يمكن للحاسب القيام بها :

$$C = A + B \quad \text{أ —}$$

$$D = A + B - D + 3 \quad \text{ب —}$$

$$E = A \div B + C \quad \text{ج —}$$

$$F = (A - B) \div 2 - C \quad \text{د —}$$

$$X = (A)^3 - (B + C) - 10 \quad \text{هـ —}$$

$$Y = [A \times B - (B)^2 \times 3] + 10 \quad \text{و —}$$

فبالنسبة لتسلسل العمليات الحسابية في العبارات السابقة فهو واضح في الثلاث الأول، وفي العبارة (د) يتم طرح B من A أولاً، ثم يقسم الناتج على ٢، وأخيراً يطرح C من الناتج الأخير. أما في المثال هـ، فيتم رفع المتغير A للقوة ٣ ثم تجمع B و C ويطرح ناتج ذلك من القيمة الناتجة عن الأس. وأخيراً يطرح ١٠ من الناتج الأخير للوصول للناتج النهائي. وأخيراً في العبارة (و) يتم ضرب الناتج من رفع B للأس ٢ في العدد ٣، وتطرح نتيجة هذه العملية من ناتج ضرب A في B، وأخيراً يضاف للناتج ١٠.

وهذا التسلسل في جزئيات العمليات الحسابية لا يتأثر بمستوى سهولة وصعوبة المعادلة الجبرية .
ولنأخذ المثالين التاليين :

$$(أ) \quad 1 - 2 \times 2 \quad (ب) \quad 1 - 2 \times 149,99$$

فبالرغم من أن الإنسان يقوم بالعملية الحسابية الأولى بسرعة ودون الحاجة إلى الرجوع للورقة والقلم، كما قد يستدعى الحال في المثال الثاني، فإن الحاسب يتبع نفس الخطوات في كلا المثالين . إذ يجب أن تخزن نتيجة الضرب أولاً، ومن ثم يتم الطرح . وبالإضافة إلى تسلسل العمليات الحسابية في المعادلة الجبرية، فإن هناك معايير

عامة تنطبق على جميع العبارات الحسابية بغض النظر عن لغة البرمجة المستخدمة :

(١) وجود متغير واحد فقط إلى يسار إشارة «=» وذلك يرمز إلى اسم الصندوق الذي سيتم فيه تخزين الناتج من العملية الحسابية .

(٢) استخدام إشارة «=» هنا ليس في معناها الجبري المجرد، فإشارة «=» تعني «إعطاء» أو «إسناد» القيمة النهائية للعمليات الحسابية إلى المتغير الظاهر على يسار الإشارة «=» .

فمعنى العبارة التالية إذن :

$$C = C + D$$

هو: أضف القيمة السابقة المخزنة في C إلى القيمة المخزنة في D وضع القيمة الناتجة في C مرة أخرى لتمثيل القيمة الجديدة لها .

(٣) استخدام رموز خاصة في لغة البرمجة لتمثيل العمليات الحسابية، وذلك كالتالى :

الرمز الجبرى المقابل	الإشارة الجبرية
+	+ الجمع
-	- الطرح
/	÷ القسمة
*	× الضرب
* * أو ↑	\times^3 الأس

ب - العمليات المنطقية : يتضمن المجالان التجارى والحكومى الكثير من التطبيقات التى تتطلب إجراء مقارنات بين بيانات مختلفة، إذ لا يخلو أى تطبيق منها . وتحديثنا فى الفصل الثانى عن التشعب المبني على مقارنة معينة وتمثيلها بالرمز التالى ◇ فى نمط البرمجة التركيبية، وسنورد هنا مثالين آخرين :

المثال الأول : إذا تصورنا الكبيئات التى تحتوى أدراجها على ملفات الموظفين لجهة حكومية معينة، فقد نرغب فى الحصول على المعلومات التالية :

- أ - عدد الموظفين الذين تزيد أعمارهم على ٤٥ سنة .
- ب - أسماء الموظفين الذين سيتقاعدون خلال السنة القادمة .
- ج - عدد الموظفين ونسبهم، حسب توزيعهم بين الجنسيات المختلفة .
- د - أسماء الموظفين التابعين لدائرة معينة موزعين حسب فئات الأعمار التالية :
أقل من ٢٠، من ٢٠ إلى أقل من ٣٠، من ٣٠ إلى أقل من ٤٠، من ٤٠ فأكثر .

- هـ — عدد الوظائف الشاغرة في السنة المالية .
- و — أسماء الموظفين التابعين لدائرة معينة وتزيد أعمارهم على ٤٠ سنة .
- ز — أسماء الموظفين التابعين للدائرة المالية أو للدائرة التجارية أو لدائرة شؤون الموظفين .
- ح — أسماء الموظفين التابعين للدائرة المالية أو للدائرة التجارية وأعمارهم تزيد على ٤٥ سنة .

المثال الثاني :

إذا تصورنا بطاقات المخزون تحتوي على معلومات عن الأنواع المختلفة للقطع المخزونة في المستودع، لأردنا الحصول على المعلومات التالية :

- (أ) قائمة برموز القطع وأوصافها، والكمية الحالية المخزنة الناقصة عن مستوى محدد مسبقاً .
- (ب) قائمة برموز القطع وأوصاف القطع، التي لا يتم الطلب عليها باستمرار .
- (ج) مجموع قيم القطع المنصرفة، موزعة حسب المناطق الجغرافية الشرقية والوسطى والغربية .
- (د) أسماء القطع المنصرفة للدائرة المالية، أو لدائرة الإنتاج وتزيد قيمتها على ١٠٠٠ ريال .
- (هـ) أسماء القطع المنصرفة للدائرة الإدارية وتزيد قيمتها على ٢٠٠٠ ريال، أو المنصرفة لدائرة شؤون الموظفين وتزيد قيمتها على ١٥٠٠ ريال .
- (و) أرقام وأسماء القطع التي يزيد سعر الواحدة منها على ٥٠٠ ريال، موزعة حسب طبيعة وحدة البيع : بالقطعة الواحدة، بـ «الدرزن»، بالرزمة .
- إن إجراء المقارنات المذكورة في المثالين السابقين يدوياً يتطلب الحفاظ على البيانات الضرورية، والمناسبة لتحقيقها في بطاقات الموظفين و بطاقات المخزون، وعند

استخدام الحاسب الآلى، يتم تخزين البيانات فى سجل لكل موظف ولكل قطعة فى المخزون . ومن الطبيعى أن تحتوى هذه السجلات على : بيانات أخرى قد لا تستخدم فى المقارنات المذكورة آنفاً .

تصنيف المقارنات : تستخدم المقارنات فى عمليات التصنيف للسجلات المخزنة فى الحاسب أو للاستخلاص الجزئى لبعض منها ، وكما مر ذكره فى بداية هذا الفصل ، فإن البيانات التى تستخدم فى المقارنات تخزن فى أسماء متغيرات (صناديق) ويطلق على كل واحدة من هذه البيانات حقل ، وتسهيلاً للشرح فى هذا الجزء ، فإن أسماء هذه المتغيرات ستكون عبارات باللغة العربية ، أما فى حالة البرمجة بإحدى اللغات ، فإن هذه الأسماء يجب أن تخضع لقوانين تلك اللغة .

ويمكن تصنيف المقارنات إلى صنفين رئيسيين :

الأول : البسيط ، ويتم فيه المقارنة بين قيمتين من البيانات فقط . وفى العادة تكون إحدى القيمتين مخزنة فى اسم متغير — العمر مثلاً — أما القيمة الأخرى فقد تكون بيانات ثابتة سواء حسابية رقمية أو غير حسابية حرفية أو بيانات متغيرة ، ومن الأمثلة على هذا الصنف : العمر < ٤٥ حيث يخزن العمر لكل موظف ويسترجع بعد ذلك لمقارنته بالرقم «٤٥» .

والكمية < الحد الأدنى

حيث يخزن الكمية والحد الأدنى لكل قطعة . وفى هذه الحالة يتم وضع الأجزاء المكونة من مقارنتين أو أكثر بين قوسين ، ويتم معاملة هذه الأجزاء كوحدة واحدة لها نتيجة خاصة بها ، ومن ثم نقارن نتائج مقارنات الأجزاء لتحديد نتيجة المقارنة الكلية ، ومن الأمثلة على ذلك :

— (اسم الدائرة «=» المالية أو اسم الدائرة «=» شؤون الموظفين) والعمر < ٤٥ .

— (اسم الدائرة = المالية والقيمة < ٢٠٠٠) أو (اسم الدائرة = «الموظفين» والقيمة < ١٥٠٠) .

ففى المثال الأول يقوم الحاسب بإجراء المقارنتين الموجودتين داخل القوسين وتحديد النتيجة لهذا الجزء، ومن ثم تربط هذه النتيجة بنتيجة المقارنة الأخيرة . وفى المقال الثانى يتم تحديد نتيجة المقارنات المجزأة والمحصورة بين الأقواس، بحيث تحدد النتيجة النهائية للعبارة بناء على النتيجتين الجزئيتين .

كيفية تنفيذ عبارة المقارنات : إن الهدف من المقارنة هو تحديد اختيار أحد مسارين بناء على تحقق شرط المقارنة أو عدمه :

— رمز المنطقة = ١

حيث يخرن رمز المنطقة و يقارن مع «١» الذى يشير إلى المنطقة الوسطى .

الثانى : المركب، حيث تحتوى العبارة الواحدة على مقارنتين، وينطبق على الصنف ما انطبق من القواعد على الصنف الأول من حيث طبيعة البيانات المقارنة، ويمكن تقسيم هذا الصنف إلى صنفين فرعيين :

١— **المركب العادى :** والذى يحتوى على المقارنات دون اعتبار لتحديد تسلسل تنفيذ أجزائها (مقارناتها) . ومن الأمثلة على هذا النوع :

— اسم الدائرة = «المالية» والعمر < ٤٥

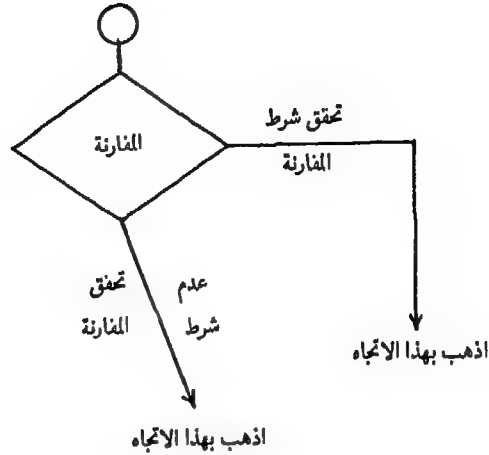
— اسم الدائرة = «المالية» والعمر < ٤٥ والحالة الاجتماعية = «أعزب» .

— اسم الدائرة = «المالية» أو اسم الدائرة = «الإدارية» أو اسم الدائرة = «الموظفين» .

— سعر الوحدة < ٥٠٠ أو اسم الدائرة المتسلسلة = «المالية» .

٢— **المركب المجزأ :** والذى يحتوى على عدة مقارنات بحيث تجمع مقارنتان أو أكثر فى جزء واحد، ومن الممكن أن تحتوى عبارة المقارنة على جزأين أو أكثر.

ويمثل الشكل التالى كيفية إجراء المقابلة :



ويمكن تبادل مكان الاتجاهين في رمز المقارنة دون أن يؤثر ذلك على سير تنفيذ الخطوات الأخرى .

الفرق بين «و» و «أو» : عند إجراء المقارنة، فإن المقارنات المركبة العادية المربوطة بـ «و» يجب أن تتحقق جميعها، حيث يتحقق الشرط الكلى للعبارة الشرطية، فمثلاً، تؤدي مقارنة اسم الدائرة «و» العمر إلى إنتاج قائمة بالأسماء التى يتحقق فيها الشرطان : أن تكون تابعة للدائرة المالية والعمر أكبر من ٤٥ ، وكذلك إذا أضيف شرط المقارنة مع الحالة الاجتماعية .

أما عند استخدام حرف «أو» فإن تحقق الشرط الكلى للعبارة يتم إذا تحققت أى من الشروط المربوطة بـ «أو» . فالمقارنات عن الدائرة المالية «أو» الدائرة الإدارية «أو» دائرة الموظفين تؤدي إلى إنتاج قائمة بأسماء الموظفين التابعين لأى من هذه الدوائر .

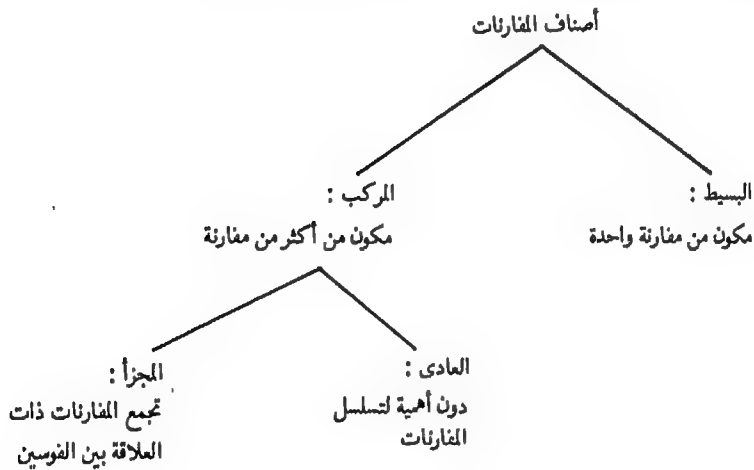
و يتبع نفس الأسلوب في إجراء المقارنات المتضمنة أقواساً، فالمقارنة عن الدائرة المالية أو الدائرة الإدارية المحصورة بين قوسين تؤدي إلى اختيار أسماء الموظفين في هاتين

الدائرتين، ومن ثم فإن مقارنة العمر تؤدي إلى حصر القائمة فقط في الموظفين الذين تزيد أعمارهم على ٤٥ عاماً .

أما المقارنة عن الدائرة المالية والقيمة أو دائرة الموظفين والقيمة، فينتج عنها أسماء القطع التي صرفت للدائرة المالية، وقيمتها تزيد على ٢٠٠٠، والقطع التي صرفت لدائرة الموظفين، وتزيد قيمتها على ١٥٠٠ .

لاحظ أن صياغة عبارة المقارنة بـ «و» عند استخدام نفس الاسم المتغير يستدعي تكراره، ومثال ذلك : إذا كان العمر < ٢٠ والعمر = أكبر من ٣٠، فنكرر استخدام العمر مرتين وليس مرة واحدة .

ويمثل الشكل الشجري التالي تلخيصاً لأصناف المقارنات :



طبيعة التخاطب مع الحاسب .:

بعد أن تحدثنا عن طبيعة البيانات وحالات معالجتها، لابد من شرح الترابط بين العمليات التي تصاحب التخاطب مع الحاسب الآلي، فالهدف الرئيسى لهذا التخاطب هو معالجة البيانات المدخلة لإنتاج المعلومات، ويتطلب ذلك تحديد خطوات المعالجة في برنامج مكتوب بإحدى لغات البرمجة . وسنستعين بالشكل التالى لشرح طبيعة التخاطب مع الحاسب عند تطوير برنامج معين .

نلاحظ من الرسم ما يلي :

١ - بعد كتابة البرنامج يتم تنفيذه باستخدام عينة من البيانات ، ويتم اختيار هذه العينة بحيث يسهل الوصول إلى النتائج يدوياً ، وذلك من أجل مقارنتها مع النتائج التى سيظهرها الحاسب ، وبحيث تغطى كافة الاحتمالات والمسارات التى يمكن أن تتخذها المعالجة .

فعند مقارنة ساعات العمل مع الرقم ٤٠ ، الذى يمثل العدد الطبيعى لساعات العمل الأسبوعية ، وذلك من أجل معرفة وجود ساعات إضافية من عدمه ، لابد أن تحتوى البيانات الاختيارية على أرقام لعدد الساعات الأسبوعية = وأكبر من وأصغر من ٤٠ . أما كون أجرة الساعة ذات أرقام عشرية ، فلا يعنى بالضرورة أن تحتوى بيانات الاختيار على أرقام مثل ٥٠,٩٩ أو ٤٩,٣٨ ، إذ أن هذه الأرقام لا تؤثر على منطق تسلسل خطوات البرنامج .

وبعد التأكد من صحة النتائج ، يتم تنفيذ البرنامج باستخدام البيانات الفعلية .

٢ - التفريق بين قائمة البرنامج الذى يحتوى على عبارات البرمجة حسب ما تتم طباعته من قبل المبرمج ، وبين النتائج التى تظهر نتيجة لتنفيذ البرنامج . ففى أى مرحلة من مراحل التخاطب مع الحاسب ، يمكن الحصول على قائمة عبارات البرنامج ، من أجل عمليات التدقيق والمراجعة بشكل أساسى ، فى البداية ، ومن أجل إرفاقها مع نتائج التنفيذ عند التأكد من تكامل وصحة المعلومات المخرجة .

٣ - يمكن التفريق بين نوعين من الأخطاء التى من الممكن أن تقع عند محاولة تنفيذ البرنامج : الأخطاء اللغوية Syntax Errors والأخطاء المنطقية Logical Errors . فالأخطاء اللغوية هى التى لها علاقة باللغة ، وتنتج عن مخالفة قواعدها . ففى لغة بيسك مثلاً ، يمكن ملاحظة الأخطاء اللغوية التالية :

10 PRNT

— (خطأ فى تهجى PRINT)

— (إسناد بيانات حرفية لمتغير رقمى) «Good Morning» = 20 A

وهناك أخطاء أخرى يمكن الرجوع إليها فى دليل اللغة المستخدم . أما الأخطاء المنطقية ، فلها علاقة بخطوات سير البرنامج ، كوصف خطوة تابعة لأخرى كان يجب أن تسبقها ، كمحاولة حساب الأجر الإضافى قبل قراءة عدد الساعات أو قبل التدقيق فيها . والأخطاء اللغوية تكتشف من قبل الحاسب عند تسلمه تعليمات اللغة . أما الأخطاء المنطقية فعلى المبرمج اكتشافها عن طريق مراجعة خطوات البرنامج ونتائج البيانات الاختيارية .

٤ — ميزات اكتشاف الأخطاء يتم حسب المرحلتين اللتين تمر بهما عملية التخاطب مع الحاسب :

— مرحلة ترجمة تعليمات اللغة إلى اللغة التى يفهمها الحاسب ، ويطلق على البرنامج الناتج عن هذه المرحلة «البرنامج الهدف» «Object Program» .
— مرحلة تنفيذ التعليمات المترجمة والتى تؤدى إلى قيام البرنامج الهدف بتنفيذ الخطوات الحسابية والمنطقية المنصوص عليها فى البرنامج . وفى المرحلة الأولى يتم اكتشاف الأخطاء اللغوية ، فى حين يتم اكتشاف الأخطاء المنطقية فى المرحلة الثانية .

ويمكن تحقيق المرحلة الأولى بأحد أسلوبين :

— أسلوب الترجمة الفورية للتعليمات (المترجم) Interpreter

— أسلوب التأليف (المؤلف) Compiler

ففى الأسلوب الأول ، تتم ترجمة أسطر البرنامج سطرأ سطرأ حال إرسالها إلى الحاسب ، ويتم اكتشاف الأخطاء اللغوية لكل سطر على حدة . ويستخدم هذا الأسلوب بشكل خاص مع لغة بيسك ، وإن لم يكن هو الأسلوب الوحيد .

أما في أسلوب التأليف، فتتم قراءة البرنامج كاملاً، ومن ثم اكتشاف الأخطاء اللغوية، بعد التدقيق في تجانس قواعد اللغة المستخدمة في البرنامج ككل متكامل. ومن الممكن استخدام هذا الأسلوب مع لغة بيسك باستخدام مؤلف اللغة بدلاً من استخدام اللغة بالأسلوب العادي المتعارف عليه، أى الأسلوب الأول.

وقد تم شرح ميزات اكتشاف الأخطاء بشكل مبسط دون الخوض في الكثير من التفاصيل.

٥ - بناءً على المناقشة السابقة، نجد أن الرسم السابق يناسب استخدام لغة بيسك مع المترجم، أما في حالة استخدام أسلوب التأليف، فيتم في العادة تنقية البرنامج من الأخطاء اللغوية قدر الإمكان، وبعد ذلك يتم دمج البرنامج مع عينة الاختبارات وتنفيذه للتأكد من خلوه من الأخطاء المنطقية.

٦ - إنه، وإن كان قد تم تنفيذ البرنامج بنجاح حسب المواصفات والمتطلبات الأولية، فلا بد وأن تطرأ ظروف ومتطلبات جديدة تتطلب إجراء التغييرات والتعديلات على البرنامج، ومن ثم يتم العودة إلى نقطة البداية والمرور في نفس المراحل المتتالية.

تمارين

١ - اشرح باختصار العمليات التي يمكن أن تتم على محتويات الصناديق (المتغيرات) في ذاكرة الحاسب الآلى .

٢ - بين كيفية تعامل الحاسب مع الجدولين التاليين :

أ - تخزين المبيعات اليومية لمدة أسبوع .

ب - تخزين المبيعات اليومية حسب موقعها من الأسابيع الأربعة في كل شهر .

٣ - ماهى أصناف البيانات حسب طبيعة الاستعمال مع إعطاء مثالين لكل صنف ؟

٤ - ماهى أصناف العبارات التوضيحية مع إعطاء نوعين لكل صنف ؟

٥ - ماهو تسلسل تنفيذ العمليات الحسابية ؟

٦ - صنف المقارنات التالية :

أ - السن أكبر من ٦٠ .

ب - الدخل أقل من ٥٠٠٠ ريال .

ج - الدارس من خارج منطقة الرياض ويسكن فى سكن المعهد .

د - الأستاذ يقوم بتدريس أكثر من عشر ساعات ويحضر للتدريس ليلاً

و يعمل فى استشارة .

هـ - المدير ذو الراتب أكثر من عشرة آلاف ريال شهرياً ، و يعمل فى فرع

الرياض ، أو المدير ذو الراتب بين خمسة آلاف إلى ثمانية آلاف ريال

و يعمل فى فرع جدة .

أساسيات استخدام لغة بيك

برمجة معالجات بسيطة.
برمجة معادلات بسيطة
مع عمليات إدخال.
الدوارة البسيطة
والتحكم بها.
تعليمات التشعب .
استخدام الدالات
في العمليات الحسابية.
التحكم في طبع البيانات
المخرجة والمعلومات.
المصفوفات ذات البعد
الواحد وتعليمات
الدوارة البسيطة.
المصفوفات ذات البعدين
ومكوناتها وتعليمات
الدوارة المتقدمة.

برمجة معالجات بسيطة

حالات التخاطب مع الحاسب الآلى :

بعد تجهيز الحاسب والوصول به إلى مرحلة قبول تعليمات لغة بيسك ، يمكن التخاطب مع الحاسب الآلى ضمن الحالات التالية :

أ) حالة الاستعداد : يمكن طباعة أى تعليمة من تعليمات لغة بيسك دون كتابة رقم سطر، فمثلا نستطيع أن نطبع التعليمات التالية :

LET A = 10

LET B = 5

LET C = A * B

PRINT A,B,C

وقد يستخدم هذا الوضع فى بداية تعلم اللغة ؛ لأخذ فكرة عامة عن كيفية تخزين البيانات فى ذاكرة الحاسب وطباعة محتوياتها ، بالإضافة إلى ذلك يمكن التوصل إلى معرفة محتويات أسماء المتغيرات بعد تنفيذ البرنامج ، كما سنتعرض له بعد قليل .

ب) حالة البرمجة : نبدأ حالة البرمجة فور طباعة رقم السطر لعبارة بيسك ، ويبين الشكل التالى مكونات العبارة :

XXXX



رقم السطر

XXXX



تعليمة

XXXXXXXX



عملية حسابية متضمنة متغيرات أو أسماء
متغيرات أو مقارنة بين متغيرات

ويجب أن يكون رقم السطر بين 00001 إلى 99999 . أما التعليمة فهي من الكلمات ذات الدالة الخاصة لترجم لغة بيسك (RESERVED WORD) ، ويجب أن تظهر بأحرفها كاملة دون زيادة أو نقصان، وسيتم كتابة هذه العبارة في الأشكال التوضيحية لها بالحرف الكبير، مثل PRINT، أما أسماء المتغيرات فسيتم استخدام الأحرف الصغيرة لها . وإذا تضمنت المفارقات رموزاً، مثل (و AND) ، (أو — OR) ، فستعامل مثل التعليمة . أما عند طباعة البرامج فنجد أن الأمر سيان في كيفية طباعة التعليمات . فمثلاً :

— الشكل التوضيحي لعبارة اطبع PRINT		
XXXXX	PRINT	X1,X2,X3
رقم السطر	تعليمة	أسماء المتغيرات
— طباعة تعليمة PRINT في برنامج معين		

10 PRINT N1,N2,N3

ولابد من الضغط على مفتاح العودة/ الإدخال بعد الانتهاء من طباعة كل عبارة .

(ج) حالة التشغيل : يتم الانتقال إلى حالة التشغيل عند طباعة كلمة RUN، ويكون ذلك بعد الانتهاء من طباعة برنامج معين إما لاختباره أو للحصول على النتائج المرجوة، كما هو في المثال التالي :

```
10 LET A = 5
20 LET B = 7
30 LET C = A + B
40 PRINT A,B,C
RUN
5      7      12
```

ويمكن بعد تنفيذ البرنامج التخاطب مع الحاسب من حالة الاستعداد . فمثلاً : بعد تنفيذ البرنامج السابق يمكن كتابة التعليمة التالية :

PRINT A,B,C

وستظهر النتائج السابقة بعد الضغط على مفتاح العودة/ الإدخال .

معالجة عمليات حسابية بسيطة :

لنبدأ بشرح أسلوب برجة الحاسب بلغة البيسك بمثال بسيط بدائي وذلك لأغراض التعليم والإيضاح .

مثال (٥ - ١) : من العمليات الحسابية الكثيرة التى يقوم بها بائع السجاد عند حساب المبلغ الإجمالى الذى سيدفعه الزبون : إيجاد مساحة السجاد المبيع . الهدف هو إيجاد مساحة السجاد المبيع إذا علم الطول والعرض .
وخطوات الحل هى :

١ - الحصول على الطول والعرض .

٢ - حساب المساحة وفقاً لمعادلة المستطيل .

المساحة = الطول \times العرض .

٣ - إيضاح النتائج .

وسنتعرض الآن للعبارات المطلوبة فى لغة البيسك ، والتى ستترجم هذه الخطوات حتى يستطيع الحاسب تنفيذها ، ونشير هنا إلى أننا سنبدأ من وضع البرمجة Program Mode والذى يتطلب إعطاء أرقام لأسطر العبارات فى لغة بيسك .

الخطوة الأولى - الحصول على البيانات المدخلة :

إن الحصول على الطول والعرض - المدخلات - اللازمة للخطوة التالية - المعالجة - يمكن التوصل إليه عن طريق أكثر من أسلوب واحد وبغض النظر عن

الأسلوب المختار، فإننا نعلم أن هذه الخطوة تتطلب تحديد صندوقين في ذاكرة الحاسب لتخزين رقمي الطول والعرض بداخلهما — متغيرين . وأسماء هذه المتغيرات الحسابية — الصناديق — تخضع لقوانين لغة بيسك، كما هي موضحة في الشكل التالي :

قواعد تسمية المتغيرات الحسابية :	
١ — يبدأ بحرف أبجدي A-Z	
٢ — لا يبدأ برقم 0-9	
٣ — يمكن أن يكون الحرف الثاني إما أبجدياً أو رقمياً .	
٤ — لا تنتهي الأسماء بإشارة \$	
A, B, BA, B1	أسماء صحيحة
AC \$, A \$, A1 \$, 1A	أسماء غير صحيحة
٥ — بعض الحاسبات تحدد عدد الحروف المستعملة باثنين أو ثلاثة أو ثلاثين .	
IBM	: ٣٠
RADIO- SHACK	: ٣
HP- 3000	: ٣٠
APPLE	: ٣
٦ — لا تحتوي على فراغ Space أو «.»	
AB, CA.1, A1 \$	فالأسماء التالية خاطئة
٧ — يمكن استخدام % للدلالة على كون الرقم المخزن كاملاً لا يحتوي على كسر عشري، مثل A %	

واستخدام أسماء المتغيرات الحسابية هذه يكون ضمن عبارات البيسك التي ستقوم بتخزين الأرقام بداخلها . ولكن النتيجة المرجوة موضحة في الشكل التالي :

ذاكرة الحاسب	
B	A
20	10

حيث يمثل الصندوق «A» العرض، والصندوق «B» الطول، والقيم ١٠، ٢٠
قدماً على التوالي * .

وللوصول إلى هذه الحالة سنتبع الآن واحداً من الأساليب المتاحة وهي استخدام
عبارة الإسناد LET

عبارة الإسناد LET :

تستخدم هذه العبارة لتخزين البيانات داخل المتغيرات – الصناديق وتتبع هذه
القاعدة العامة .

أحد أشكال عبارة الإسناد LET				
رقم السطر	(ثابت)	متغير	(ثابت)	رقم
XXX	↓ LET	XXX	↓ =	XXX

وعلى ذلك تكون العبارة المطلوبة لتخزين الطول والعرض في الصناديق المحددة
كالتالى، مع إعطاء أرقام متتالية لهذه العبارات حسب قواعد بيسك :

ترجمة الخطوة الأولى إلى عبارات بلغة بيسك	
١٠ LET A = 10	١ – الحصول على الطول والعرض
20 LET B = 20	(البيانات المدخلة)

الخطوة الثانية – (حساب المساحة – المعالجة) :

إن حساب المساحة يتطلب إسناداً من شكل آخر، فبدلاً من إسناد قيمة رقمية
معروفة مسبقاً، فإن القيمة التى ستخزن فى هذه الحالة ستكون نتيجة لعملية حسابية
هى الضرب .

* يتم التخزين داخل الحاسب للأرقام باللغة الإنجليزية، وسنشير فيما بعد إلى طريقة لتحويل الأرقام الإنجليزية إلى
عربية بهدف طباعتها .

وتتطلب هذه الخطوة تحديد اسم لمتغير جديد هو المساحة ، والوضع المطلوب هو

التالى :

ذاكرة الحاسب		
C	B	A
200	20	10

حيث يمثل المتغير «C» المساحة ، وكما هو الحال فى الخطوة الأولى ، تستخدم هذه الصناديق داخل عبارة الإسناد ، كما هو فى الشكل التالى :

أحد أشكال عبارة الإسناد LET			
رقم السطر	متغير - ١	متغير - ٢	متغير - ٣
↓	↓	↓	↓
XXX	LET XXX	= XXXX	○ XXXX

وعليه تكون ترجمة الخطوة الثالثة كالتالى :

ترجمة الخطوة الثالثة بلغة بيسك
٢ - حساب المساحة (المعالجة)
30 LET C = A ○ B

الخطوة الثالثة - (إيضاح الناتج - المخرجات) :

إيضاح الناتج قد يتم بوسائل متعددة ، والمشهور منها والمتبع فى لغة بيسك فى كثير من الأحيان ، هو فى إيضاح الناتج على شاشة العرض (CRT) والعبارة المستعملة هى PRINT حسب الشكل التالى :

أحد أشكال عبارة الطباعة PRINT		
رقم السطر	متغير - ١	
XXX	PRINT	XXX

وبناء على ذلك تكون ترجمة الخطوة الثالثة كما هو موضح بالشكل التالى :

ترجمة الخطوة الثالثة بلغة بيسك
٣- إيفضاح الناتج (المخرجات)
40 PRINT C

و يبين الشكل التالى خطوات المعالجة وترجمتها إلى عبارات بلغة بيسك :

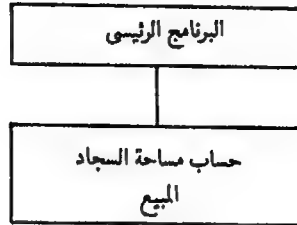
عبارات بيسك	الهدف : حساب مساحة السجاد
10 LET A = 10	١- الحصول على البيانات
20 LET B = 20	الطول والعرض .
30 LET C = A × B	٢- حساب المساحة (المعالجة)
40 PRINT C	٣- إيفضاح الناتج (المخرجات)

ويمكنك طباعة البرنامج البسيط هذا، والانتقال إلى وضع التنفيذ Execution Mode عن طريق طبع عبارة RUN ومن ثم الضغط على مفتاح الإدخال ENTER ، وستظهر النتيجة أمامك على الشاشة .

و ملخص العبارات التى تم التعرض لها إلى هذه المرحلة موضحة فى الشكل التالى :

ملخص عبارات البيسك					
XXX	LET	XXX	=	XXX	عبارة الإسناد
		متغير - ١		قيمة عددية	
XXX	LET	XXX	=	XXX	متغير - ٣
		متغير - ١		متغير - ٢	متغير - ٣
XXX	LET	XXX	=	XXX	XXXX
		متغير			
XXX	PRINT	XXX			عبارة الإخراج

اتباع أسلوب الهيكل الهرمى فى التخطيط للبرنامج وأساليب البرمجة الهيكلية :
 إن الهيكل الهرمى لهذا البرنامج يمكن تصوره وكأنه ذو جزء واحد تنفيذى ، له
 وظيفة حساب مساحة السجاد المبيع ، ويعكس الرسم التالى الهيكل الهرمى البدائى :



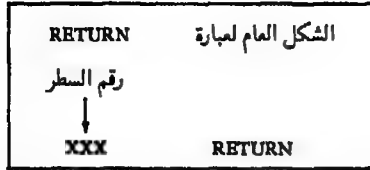
فتشكل العبارات الأربع البرنامج الخاص بالجزء ، وعلينا أن نكتب البرنامج الرئيسى الذى سيتحكم فى الجزء التنفيذى . وبعمل ذلك لابد من شرح العبارة التى ستقوم بإشعار الحاسب بالذهاب إلى الجزء المنفذ ، والعودة منه عقب الانتهاء إلى الجزء المتحكم .

تعليمية الذهاب إلى جزء منفذ **GOSUB** : تقوم هذه التعليمة بتحويل مسار تنفيذ البرنامج من الجزء المتحكم إلى الجزء التنفيذى ، ولها الشكل التالى :

أحد أشكال تعليمة GOSUB	
رقم السطر الذى يبدأ به الجزء المنفذ	
	↓
GOSUB	XXX

ولابد أن يحتوى الجزء المنفذ لتعليمة **RETURN** فى نهايته ؛ لإشعار الحاسب بانتهاء تنفيذ الجزء والعودة إلى الجزء المتحكم ، أى إلى السطر الذى يلي السطر الذى وجدت به تعليمة **GOSUB** الخاصة بهذا الجزء .

والشكل العام لهذه التعليمة موضح كالتالى :



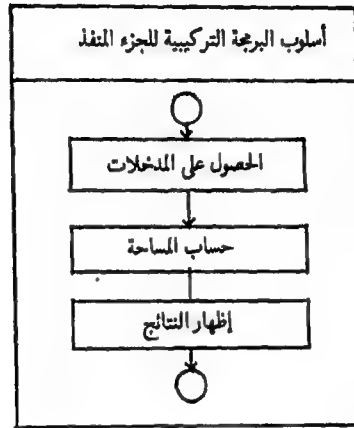
وفي حالة وجود برنامج رئيسي متحكم وآخر أو أكثر تنفيذي، لابد من تحديد نهاية كل جزء باستخدام عبارة END، لذلك فإن هذه العبارة تظهر بعدد الأجزاء الرئيسية والفرعية، في نهاية الأسطر الخاصة بكل جزء. ويكون البرنامج للمثال قيد الشرح كالتالي :

شكل (٥ - ١)

استخدام الهيكل الهرمي لكتابة برنامج لحساب مساحة السجاد المبيع	
10 REM	البرنامج الرئيسي المتحكم
20 GOSUB 100	
30 END	
100 REM	بداية البرنامج الفرعي المنفذ
110 LET A = 10	
120 LET A = 20	
130 LET C = A * B	
140 PRINT C	
150 RETURN	

لاحظ أننا تعرضنا لتعليمة REM وهي خاصة بإضافة عبارات توضيحية تتخلل البرنامج، وليس لها أى تأثير على سير تنفيذ البرنامج إذ أن الحاسب يتجاهلها تماما عند الوصول إليها، ويستمر في تنفيذ البرامج.

ولاحظ أيضا إعادة ترقيم سطور البرنامج الفرعى المنفذ فعند الوصول إلى سطر رقم 150 يعود مسار تنفيذ البرنامج إلى السطر 30 .
أما عن تصنيف أسلوب كتابة البرنامج الفرعى وفق البرمجة التركيبية ، فهو من النوع المتتالى SEQUENCE : كما هو موضح كالتالى :



أشكال أخرى للتعليمات المشروحة :

تعليمية الإسناد LET : يمكن أن تحتوى تعليمة الإسناد على متغيرات (أسماء صناديق) كثيرة بالإضافة إلى أرقام ثابتة ، كالقسمة على ٢ فى حالة حساب مساحة المثلث مثلاً .

بالإضافة إلى ذلك يمكن أن تحتوى أيضا على العمليات الحسابية كلها من ضرب وقسمة وجمع وطرح وأس . وتعكس الأمثلة التالية هذه الأشكال من استعمال تعليمية الإسناد .

أشكال تعليمة الإسناد LET
$xxx \text{ LET } X1 = X2 + X3 / X4$
$xxx \text{ LET } X5 = X3 / X4 - X5 * X6$
حرف X متبوع برقم اسم لمتغير (صندوق) xxx رقم السطر

مثال : المطلوب : حساب ثمن السجاد المبيع إذا علم الطول والعرض بالأقدام ،
وسعر القدم المربع ، فتكون صيغة العبارة الآن كالتالى :

$$130 \text{ LET } C = A * B * P$$

حيث يمثل اسم المتغير P سعر القدم المربع . ولا بد من إضافة عملية الإسناد
للسعر :

$$125 \text{ LET } P = 5$$

لاحظ استخدام السطر رقم 125 غير الموجود فى البرنامج السابق لإضافة التعليمة
الجديدة ، فى حين تم استخدام رقم السطر القديم 130 لحساب ثمن السجاد . وعند
طباعة هذين السطرين فليس هناك حاجة إلى طباعة أى من الأسطر الأخرى القديمة ،
إذ أن الحاسب إما أن يضيف الأسطر الجديدة أو يستبدلها ، وذلك حسب الأرقام
المعطاة لها . ويتم ذلك تلقائياً دون تدخل المبرمج وحسب تسلسل أرقام الأسطر فى
البرنامج ككل .

مثال : الهدف : حساب مساحة الدائرة إذا علم نصف قطرها حسب المعادلة :

$$\text{المساحة} = \text{نق}^2 ط$$

حيث إن ط = ٣,١٤١٥٩ ، نق = نصف القطر .

خطوات الحل	عبارات بيسك
١ - الحصول على نصف القطر	110 LET N = 10
٢ - حساب مساحة الدائرة	120 LET M = N ↑ 2 * 3.14159
٣ - إيضاح النتائج على الشاشة	130 PRINT M

وفي حالة تعدد العمليات الحسابية في عبارة الإسناد LET ، فإن تسلسل تنفيذها يخضع لقواعد لغات البرمجة كما بينت في الفصل الرابع ، ونعيدها هنا كمراجعة .

تسلسل العمليات الحسابية في عبارة الإسناد LET	
١ - ↑	الرفع إلى قوة (أس)
٢ - *	الضرب والقسمة
٣ - +	الطرح والجمع

ففي المثال السابق يتم رفع الرقم ١٠ للأس ٢ ومن ثم يضرب الناتج بالنسبة التقريبية . وفي حالة وجود عمليات حسابية متشابهة ، فإن التنفيذ لها يبدأ من بعد إشارة «=» و يتجه إلى نهاية العبارة . ففي المثال التالي حيث إن $K = 4, d = 3, C = 6, B = 5$

$$1010 \text{ LET } A = B * C - D * K + 3$$

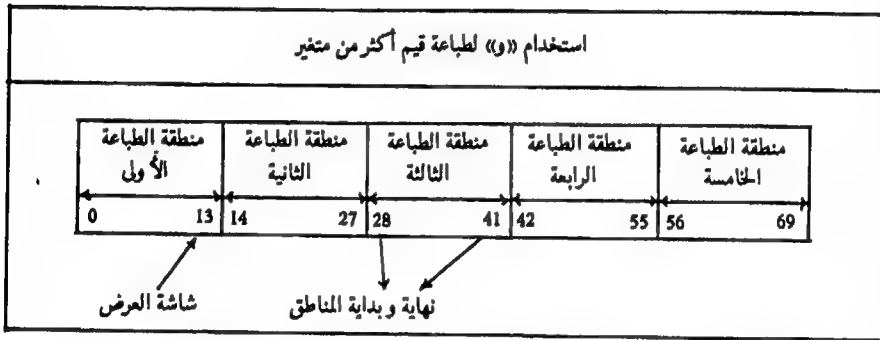
يتم التنفيذ حسب التسلسل الموضح التالي :

- | | |
|------------------|----------------------------|
| $5 * 6 = 30$ | ١ - عملية الضرب الأولى |
| $30 * 6 = 180$ | ٢ - عملية الضرب الثانية |
| $180 - 12 = 168$ | ٣ - إشارة الطرح - |
| $168 + 3 = 171$ | ٤ - إشارة الجمع + |
| | ٥ - يحزن الناتج 171 في A . |

تعليمات وأساليب إيضاح المخرجات على شاشة العرض **PRINT; TAB;** : تم استخدام هذه التعليمات لإيضاح نتائج المعالجة لمتغير واحد - أى محتويات صندوق واحد - ولكن قد يكون المطلوب إظهار محتويات أكثر من صندوق ، ففي الحالة هذه يكون استخدام تعليمة **PRINT** كالتالي :

أحد أشكال تعليمة PRINT لأكثر من متغير		
رقم السطر	متغيرات	
X X X	PRINT	X1, X2, X3

وتحدد قوانين لغة بيسك شكل طباعة هذه المخرجات على شاشة العرض . فالسطر على الشاشة مقسم إلى خمس مناطق طباعة PRINTING ZONES ، كل واحدة منها مؤلفة من ١٤ حرفاً . لذلك فإن كل سطر يمكن أن يطبع فيه قيم خمسة متغيرات ، كما يوضحه الشكل التالي :



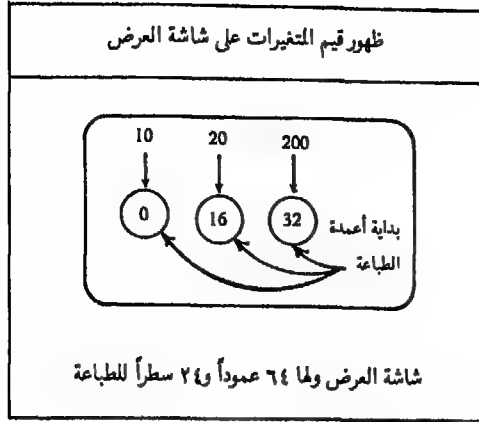
ه على جهاز IBM/PC ، وتفاوت على الأجهزة الأخرى ، ومعظمها ١٦ حرفاً مع أربع مناطق .

فلو أردنا طباعة الطول والعرض بالإضافة إلى المساحة كما في السطر رقم 140 شكل ١ ستصبح كالتالي :

140 PRINT A, R, C

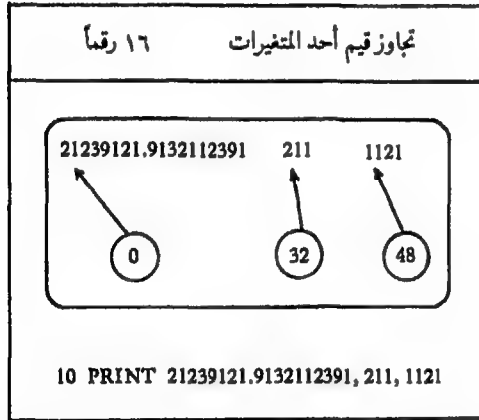
فتظهر قيم المتغيرات متسلسلة من الشمال إلى اليمين كتسلسل ظهورها في عبارة الـ

PRINT



ويخضع استخدام «و» مع تعليمة PRINT للقواعد التالية :

١ - إذا زاد عدد الأرقام المطبوعة لقيمة متغير واحد على ١٦ حرفاً ، فإن قيمة المتغير التالي ستطبع في المنطقة التي تلي الحالية باثنتين . و يوضح ذلك الشكل التالي :



٢ - في حال تخطى عملية الطباعة المناطق الخمس لكل سطر ، يتم الانتقال إلى السطر التالي تلقائياً . ففي العبارة التالية تطبع قيمة W في بداية السطر التالي :

15 PRINT P, C, A, D, W, Z

استخدم الفاصلة المنقوطة و TAB : ربما يود المبرمج ترتيب المسافات بين أرقام المتغيرات المطبوعة على الشاشة على غير ما تمليه قواعد الفاصلة «;». فمن الممكن استخدام «;» بدلاً من «;» لطباعة الأرقام دون أية مسافات فيما بينها — إلا مسافة عمود واحد يتركه الحاسب بين القيم الرقمية عند إظهارها على الشاشة.

ففى نفس المثال السابق ، يظهر السطر 140 على الشكل التالى :

140 PRINT A;B;C

وعليه فإن الشكل العام لاستخدام «;» يكون كالتالى :

الشكل العام لاستخدام PRINT مع		
المتغيرات الرقمية	رقم السطر	
X1; X2; X3;	PRINT	X X X

أما TAB فتستخدم للتحكم فى المسافات بين قيم المتغيرات — الصناديق — المطبوعة . ويكون الشكل العام لها كالتالى :

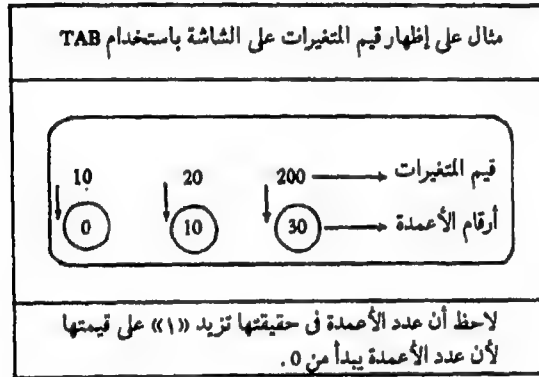
الشكل العام لاستخدام TAB مع PRINT		
المتغيرات	رقم السطر	
X1; X2	PRINT	X X X
عدد الأعمدة (المسافة) من الهامش الأيسر إلى مكان إظهار أول رقم من قيم المتغيرات		
تأكد من عدم ترك مسافة بين حرف B وإشارة «(»		

و يتم تحديد قيمتي Y1 و Y2 عن طريق معرفة عدد الأعمدة المرغوب في تركها إلى يسار القيمة المظهرة على الشاشة ، ونلاحظ أن قيمة Y2 ، وهي لاحقة وأكبر من قيمة Y1 ، يعود تحديدها إلى العمود الصفر في بداية الشاشة أيضاً .

ففى المثال السابق ، يمكن طباعة قيم المتغيرات A , B , C متباعدة بعضها عن بعض حسب رغبة المبرمج . ويمكن للسطر 140 أن يصبح كالتالى :

```
140 PRINT A; TAB (10); B; TAB (30); C
```

وتظهر النتيجة كالتالى :



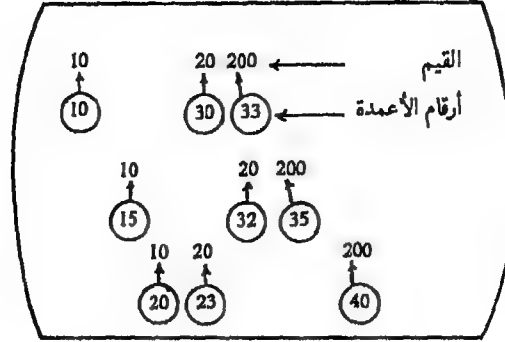
كذلك يمكن استخدام أى من «,» و «;» و «TAB» بعضها مع بعض فى سطر واحد ، كالأمثلة التالية للسطر 140 :

```
140 PRINT TAB (10); A; TAB (30); B, C
```

```
140 PRINT TAB (15); A, B; C
```

```
140 PRINT TAB (20); A; B; TAB (40); C
```

وتظهر النتائج كالتالى :



مثال متكامل لمعالجة عملية حسابية بسيطة :

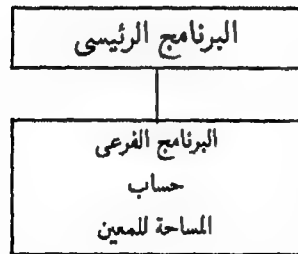
مثال (٥ - ٢) :

الهدف : حساب مساحة المعين إذا علم قطراه. تتضمن مراحل الحل المتكاملة الخطوات الرئيسية التالية* :

أولاً - خطوات الحل :

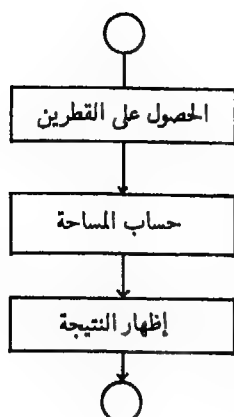
- ١ - الحصول على قيمة القطرين .
- ٢ - المساحة = القطر الأول \times القطر الثانى / ٢ .
- ٣ - إظهار الناتج .

ثانياً - الهيكل الهرمى :



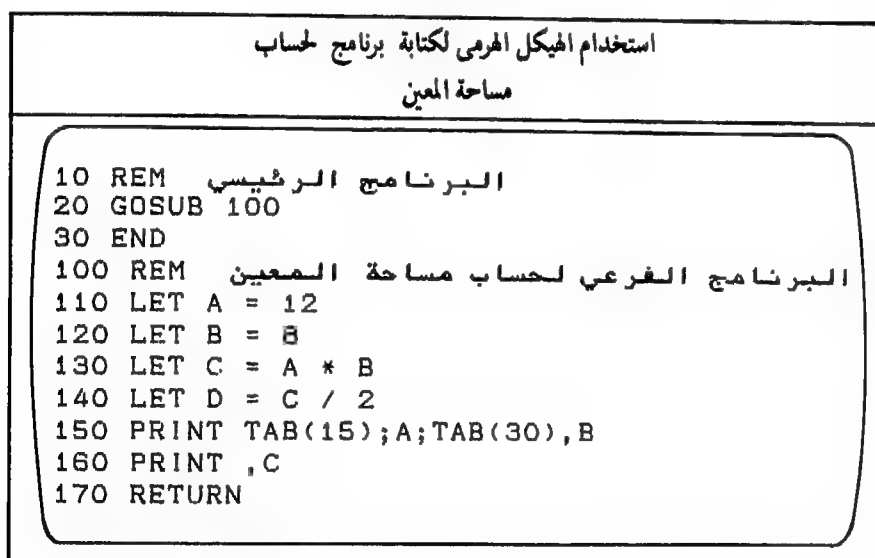
* سيتم اتباع هذه الخطوات الرئيسية الأربع فى الأمثلة القادمة .

ثالثاً - شكل البرمجة الهيكلية :



رابعاً - البرنامج :

شكل (٥ - ٢)



ملاحظات على البرنامج :

- ١ - توزيع عمليات الإسناد على سطرين بدلاً من جمعها في سطر واحد (120, 110) .
- ٢ - توزيع العمليات الحسابية على سطري الإسناد 140, 130 .

٣ - توزيع عمليات إظهار النتائج في سطرى الطباعة على شاشة العرض 150, 160 .
 وفى سطر 160 تتم طباعة قيمة المتغير «C» ابتداء من العمود ١٦ نظراً لوجود «»
 قبلها ويمكن تكرار «» . وكذلك إذا وضعت «» فى نهاية السطر 150 ، فإن قيمة
 C تظهر فى نفس السطر ، إذا كان هناك مساحة كافية ، وإلا فتظهر فى السطر
 التالى .

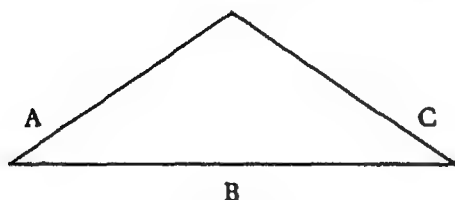
ويمكن للقارئ إعادة كتابة عبارات الطباعة على شاشة العرض باستخدام «;» و
 TAB للتمرين الثلاثة السابقة .

معالجة عمليات حسابية مع أقواس :

سنقدم الآن مثلاً عن استخدام الأقواس فى العمليات الحسابية فى عملية الإسناد
 إضافة إلى العمليات الحسابية الأخرى :

مثال (٥ - ٣) :

الهدف : إيجاد مساحة ومحيط مثلث إذا علم أطوال أضلاعه .



ملاحظة :

المحيط = مجموع أطوال الأضلاع الثلاثة

$$C + B + A =$$

$$\sqrt{D (D - C) (D - B) (D - A)} = \text{المساحة}$$

$$\frac{C + B + A}{2} = \text{حيث إن } D$$

أولاً - خطوات الحل :

١ - الحصول على أطوال الأضلاع عن طريق الإسناد

$$10 \text{ LET } A = 5$$

$$20 \text{ LET } B = 8$$

$$30 \text{ LET } C = 7$$

٢ - حساب المحيط : $100 \text{ LET } E = A + B + C$

٣ - إيجاد المساحة :

$$200 \text{ LET } D = (A + B + C) / 2$$
 إيجاد قيمة D :

ب - نوجد الفرق بين كل من الأطوال الثلاثة وقيمة المتغير D .

ج - نضرب ناتج الفرق بين كل من الأطوال الثلاثة بعضها ببعض ، ومن ثم نضرب الناتج في قيمة المتغير D

$$210 \text{ LET } F = (D - A) * (D - B) * (D - C)$$

$$220 \text{ LET } J = F * D$$

يمكن اختصار العمليتين بعملية واحدة كالتالى :

$$210 \text{ LET } F = D * (D - A) * (D - B) * (D - C)$$

ومن ثم إيجاد الجذر التربيعى ، ويمكننا عمل ذلك برفع قيمة المتغير F للقوة الثابتة 0.5 كالتالى :

$$220 \text{ LET } F = F \uparrow .5$$

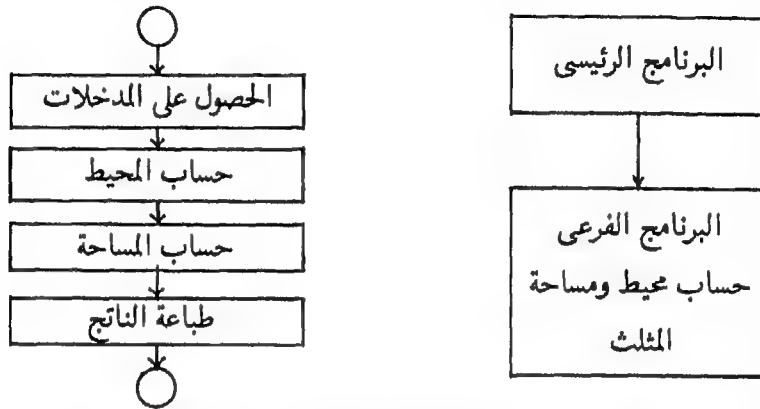
٤ - طباعة طول كل من الأضلاع ثم المساحة ثم المحيط .

$$250 \text{ PRINT } A, B, C, F, E$$

لاحظ هنا وجود الأقواس ، أى أنه يجب حساب مجموع أطوال الأضلاع الثلاثة قبل أن نقسم على 2 . وإلا تمت قسمة القيمة التى يحملها المتغير C على 2 أولاً ومن ثم

عمليات الجمع ، وهذا يوصلنا إلى نتيجة خاطئة . أى أن معالجة العمليات الحسابية المعزولة بين الأقواس أولاً ، ومن ثم تتم العمليات الحسابية الأخرى حسب التسلسل المشروح سابقاً .

ثانياً - الهيكل الهرمى للبرنامج :



رابعاً : ويتضمن الشكل التالى برنامجاً كاملاً وفق أساليب البرمجة الحديثة .

شكل (٥ - ٣)

برنامج لحساب محيط ومساحة أى مثلث إذا علم أطوال أضلاعه	
10 REM	البرنامج الرئيسى
20 REM	يعمل هذا البرنامج على ايجاد مساحة و محيط مثلث
30 REM	و من كم ايجاد المحيط و المساحة
40 GOSUB	150
50 END	
150 REM	للمحصل على طول كل ضلع بواسطة تعليمة اسند
160 LET	A = 5
170 LET	B = 8
180 LET	C = 7
190 REM	لايجاد المحيط
200 LET	E = A + B + C
300 REM	يجاد المساحة
310 LET	D = (A + B + C) / 2
320 LET	F = (D * (D - A) * (D - B) * (D - C)) ^ .5
400 REM	طباعة اخراج النتائج
410 PRINT	TAB(5);A;TAB(12);B;TAB(19);C;TAB(25);F;TAB(37);E
420 RETURN	

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذ البرامج حسب تسلسل هذه التعديلات .

٢ - بين الأخطاء إن وجدت في القيم العددية الثابتة التالية :

521,403	254.71	\$405.50	\$711.53
31-415	17.705.3	875.80	220.71

٣ - بين الأخطاء إن وجدت في اختيار أسماء المتغيرات العددية التالية :

A	AS	A 102	A \$ 2
NAME	A + 1	B%	A.7

٤ - ماهي نتائج التعبيرات الحسابية التالية (مع الأخذ بعين الاعتبار بأولوية التنفيذ) إذا كانت قيمة كل من : $A = 3$, $B = 4$, $C = 2$,

— $A / B - C$

— $A + B * B / C + 2$

— $4 + (B - C) * A - 7$

— $A / B / C * * 2$

— $(A + B) * (C - 2) * * 4$

٥ - ماهي نتائج المعادلات الجبرية التالية إذا كانت قيم كل من :

$C = 1$, $B = 2$, $A = 4$

— $A + B - C$

— $-(A + B)$

— $(A \times B)^2 + C$

$$- A^2 + AB - C^3$$

$$- \left(\frac{A}{B} + B \right)^2 - \left(\frac{C}{B} \right) + 4$$

٦ - اعمل على تحويل المعادلات الجبرية التالية إلى تعبيرات حسابية :

$$- (A \times B) + (C - B)$$

$$- A^2 + AB - C^3$$

$$- \frac{(A + B)(A - B)}{(A + C)^2}$$

$$- A^3 - 15.5 + 7B - C$$

$$- (AB^2 + BC) B$$

٧ - بين الأخطاء إن وجدت في عبارات الإسناد التالية مع تعليل السبب :

10	LET	A = 70
10	LET	A + B = C
10	LET	A = « 'ALT »
10	LET	A = (5 + B) - 7 + * 8
10	LET	A \$ = 15 - C
10	LET	A = A + 2
10	LET	A = « 3 + 5 »
10	LET	A + B = B + A
10	LET	A + B - 7

٨ - بين نتائج تنفيذ العبارات التالية :

10	PRINT	10 , 15 , 20
10	PRINT	TAB (10) , 50 , TAB (30) , 60

10	PRINT	30 , 60 ; 70 , 100
10	PRINT	37 . 512 , 55 , 60 * 2 , 31 - 15 , 12
10	PRINT	TAB (50) , 17 , TAB (50) ; 25
10	PRINT	TAB (17) , 51 . 99 ; TAB (50) ; 91

٩ - إحدى الشركات المنفذة للمشاريع الصناعية تقوم بحساب التكلفة النهائية للمشروع كالتالى :

١ - تكاليف الأيدى العاملة = عدد الساعات x أجرة الساعة .

٢ - تكاليف الآلات = عدد الساعات x أجرة الساعة .

٣ - تكاليف المواد .

٤ - التكاليف العارضة ١٠% من مجموع تكاليف الأيدى العاملة والآلات .

٥ - التكاليف الإدارية المتعددة = ٥% من مجموع التكاليف السابقة كلها .

طور برنامجاً لحساب مجموع التكاليف للمشروع .

١٠ - تتسلم إدارة المشاريع والصيانة التابعة لوزارة الصحة طلبيات الصيانة والتي قد تتطلب القيام بوحدة أو أكثر من العمليات التالية :

الدهان ، الكهرباء ، السباكة ، النجارة ، البناء ، وبعد الانتهاء من

العمليات لكل طلبية يعبأ على نفس الطلبية تكلفة كل عملية بالريالات .

اكتب برنامجاً لحساب مجموع التكلفة لكل طلبية .

برمجة معادلات بسيطة مع عمليات إدخال

مقدمة عن عمليات الإدخال إلى الحاسب :

قدمنا فيما سبق شرحاً لأحد أساليب تخزين البيانات في ذاكرة الحاسب ، وذلك باستخدام تعليمة الإسناد LET . ولكن هذا الأسلوب يشوبه بعض العقبات والصعاب التي تجعل استخدامه لغرض تخزين بيانات الإدخال عملية عقيمة .

ولنعد إلى أحد الأمثلة السابقة ، وهي حساب مساحة السجاد ، فكانت خطوات الحل والتعليمات المترجمة لها بلغة بيسك كما يلي (ونعيدها هنا للمراجعة والإشارة إليها) :

الهدف : حساب مساحة السجاد المبيع إذا علم الطول والعرض :

خطوات الحل	العبارات المقابلة
١ — الحصول على الطول والعرض	10 LET A = 10
٢ — حساب مساحة السجاد	20 LET B = 20
٣ — طباعة الناتج على الشاشة	30 LET C = A * B
	40 PRINT C

يتم طباعة البرنامج وتنفيذه للزبون الأول ، وإذا قام زبون آخر بشراء سجاد من هذا المحل ، فيجب تغيير السطرين 10 ، 20 وإدخال رقمي الطول والعرض للزبون الجديد ، وهكذا مع كل بيع للسجاد . وهذا بطبيعة الحال ليس عملياً . ومحدودية هذا الاستعمال لتعليمة LET تمنع استخدامها لهذا الغرض .

إذن فلا بد من وجود وسائل أخرى لإدخال البيانات إلى أسماء المتغيرات (الصناديق) دون الحاجة إلى تغيير أسطر التعليمات في البرنامج .

وهذا الجهد الأخير يجب الاكتفاء منه بأقل القليل ؛ فالبرنامج إذا حقق هدفه ، فإن تغييره دون سبب جوهري يؤدي إلى مضیعة في الوقت والجهد ، ويسبب الإرباك للمستفيدين من البرنامج .

وتتبع أساليب إدخال البيانات أحد الإجراءين التاليين :

١ - تخزين البيانات المدخلة كلها ، ومن ثم جعل البرنامج يطلبها واحدة تلو الأخرى حسب طبيعة المشكلة ، وهذا ما يسمى بالمعالجة الدفعية Batch Processing . وهذه البيانات في برنامج مكتوب بلغة بيسك يمكن أن تخزن داخل البرنامج نفسه ، سواء في آخره أم متناثرة خلاله ، حسب طبيعة تنفيذ البرنامج .

٢ - إدخال البيانات مفرزة حسب طبيعة المشكلة من لوحة المفاتيح ، ومن ثم معالجتها قبل إدخال ما تليها من البيانات ، وهذا يطلق عليه Transaction Processing .

ففى المثال السابق ، إما أن تخزن أطوال وأعراض السجاد المبيع لكل الزبائن ، ومن ثم تتم معالجتها معاً ، أو يدخل الطول والعرض لكل زبون ، ومن ثم تحسب المساحة وتطبع على الشاشة ، وبعد ذلك يتم إدخال المبيع الآخر ومعالجته بشكل إفرادى إلى حين الانتهاء منه .

وسنتعرض في هذا الفصل إلى كيفية استخدام عبارات لغة بيسك وفق هذين الأسلوبين ولمجموعة واحدة من البيانات ، وسنكمل النقاش في الفصل القادم لمعالجة مجموعات من البيانات .

تطوير برنامج لحل مشكلة حسابية باستخدام تعليمتى READ / DATA :

تسمح عبارة READ بتخزين البيانات المدخلة متتالية خلال البرنامج . ويشترط أن يصاحب استخدام هذه العبارة ، طبع البيانات التابعة لها على أسطر تبدأ بعبارة DATA وذلك حتى تأخذ صبغة مميزة وفريدة مقارنة بالتعليمات الأخرى . وفيما يلي مثال على استخدام هاتين العبارتين :

مثال (٦ - ١) :

الهدف : حساب العمولة لمندوبى المبيعات إذا علمت النسبة ومجموع المبيعات .

أولاً - خطوات الحل :

١ - الحصول على النسبة ومجموع المبيعات .

٢ - حساب العمولة .

٣ - إظهار المخرجات على الشاشة .

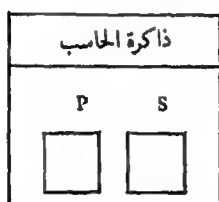
إن الفرق بين هذه الخطوات والتي سبقتها هو في ترجمة الخطوة الأولى . فبدلاً من استخدام أسلوب الإسناد سنستخدم أسلوب إدراج البيانات داخل البرنامج . والشكل العام لاستخدام عبارة READ كما يلي :

الشكل العام لعبارة READ	
X X X	READ X 1, X 2, X 3
↓	↓
رقم السطر	أسماء المتغيرات

و يلاحظ أن عدد أسماء المتغيرات التي تظهر بعد كلمة READ يتأثر بعدد البيانات المدخلة . وفي مثالنا هذا ، نحتاج إلى صندوقين داخل الذاكرة : واحد للعمولة ، وآخر للمبيعات ، وعليه فإن العبارة تكون في شكلها النهائي :

10 READ P , S

حيث يرمز P إلى العمولة و S للمبيعات بالريال .
أما في الذاكرة ، فتكون هذه المتغيرات كالتالي :



وفي حالة تنفيذ هذه العبارة ، يبحث البرنامج عن أول سطر تظهر فيه عبارة DATA ، وتخزن اثنتان من البيانات يفصل بينهما «،» .
ثانياً وثالثاً - يشبه الهيكل الهرمي ونمط البرمجة التركيبية : المثال الأخير في الفصل السابق . رابعاً - ويمثل الشكل التالي البرنامج كاملاً وفق الأساليب الحديثة في البرمجة .

شكل (٦ - ١)

برنامج لحساب العمولة لتدوين المبيعات	
10	REM البرنامج الرئيسي
20	GOSUB 100
30	END
100	REM برنامج فرعي لحساب العمولة
110	REM لقراءة العمولة والمبيعات
120	READ P, S
130	LET M = P * S
140	REM طباعة نسبة العمولة، المبيعات و العمولة
150	PRINT P, S, M
160	DATA .10, 10000
170	RETURN

ملاحظات على البرنامج :

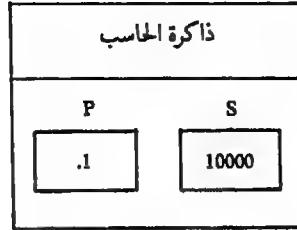
١ - تكون محتويات المتغيرات (الصناديق) على الشكل التالي :

ذاكرة الحاسب الآلى		
P	S	M
.10	10000	100

٢ - من الممكن أن تظهر عبارتا READ / DATA على النحو التالى :

أشكال من اقتران READ مع DATA	
١	٤
105 READ P 110 READ S 140 DATA .1,10000	110 READ P , S . . . 140 DATA .1 150 DATA 10000
٢	٥
(مع إلغاء 140) 105 READ P 107 DATA .1 110 READ S 112 DATA 10000	(مع إلغاء 140) 105 DATA .1,000 110 READ P, S
٣	٦
105 READ P 110 READ S 140 DATA .1 150 DATA 10000	105 READ P, S 140 DATA .1, 10000

وفي كل هذه الأشكال تكون نتيجة تنفيذ هذه العبارات تخزين القيم 1 ، 10000 في ذاكرة الحاسب على النحو التالي :



وذلك لأن تخزين قيمتي المتغيرين S, P يتم عن طريق ترتيب ظهور هذين المتغيرين في تعليمة READ ، وظهور قيمتيهما الموابتين لهما في تعليمة DATA . لاحظ في الأسلوب رقم ٦ أن ظهور « » في نهاية سطر DATA تعنى استمرار البيانات في السطر الذي يليه دون الحاجة إلى طباعة رقم السطر وكلمة DATA .

فالمبرمج عليه الحرص دائماً على موافقة ترتيب أسماء المتغيرات — الصناديق — مع قيمها المطابقة لها ، وإلا فإن الحاسب لن يكتشف الخطأ في التسلسل .
وسنقدم الآن مثلاً آخر لاستخدام عبارتي READ / DATA ، وإعطاء أمثلة عن بعض الأخطاء التي يقع فيها المبتدئون في أكثر الأحيان .

مثال (٦ - ٢) :

الهدف : إيجاد المعدل العام لعلامات دارس تقدم إلى امتحان قبول مكون من خمسة أجزاء .

أولاً - خطوات الحل :

١ - قراءة علامات الأجزاء الخمسة .

٢ - إيجاد المجموع .

٣ - إيجاد المعدل العام .

٤ - طباعة العلامات الخمس والمعدل العام .

ثانياً وثالثاً - كما في المثال السابق .

رابعاً - الحل كاملاً لهذا المثال حسب الأساليب الحديثة في البرمجة .

شكل (٦ - ٢)

برنامج لحساب معدل خمس علامات	
10 REM	استدعاء برنامج فرعي لحساب المعدل العام و طباعة النتائج
20 GOSUB 100	
30 END	
100 REM	لقراءة الدرجات الخمسة و إيجاد المعدل العام و طباعته
110 READ T1,T2,T3,T4,T5	
120 DATA 75,85,90,80,70	
130 LET A = (T1 + T2 + T3 + T4 + T5) / 5	
140 PRINT TAB(5);T1;TAB(10);T2;TAB(15);T3;TAB(20);T4;TAB(25);T5;TAB(30);A	
150 RETURN	

وستظهر ذاكرة الحاسب بعد تنفيذ هذا البرنامج كالتالى :

شكل الذاكرة بعد قراءة العلامات الخمس				
T1	T2	T3	T4	T5
75	85	90	80	70

ومن الأخطاء الممكن حدوثها هي :

١ - نقصان عدد القيم في عبارة DATA عن عدد أسماء المتغيرات في عبارة READ . ففي المثال السابق لنفرض أن المبرمج نسى أن يطبع القيمة الأخيرة (70) في عبارة DATA في السطر 80 ، بحيث ظهر سطر 80 كالتالي :

80 DATA 75 , 85 , 90 , 80

ففي هذه الحالة يتوقف تنفيذ البرنامج مشيراً إلى وجود نقص في البيانات .

٢ - زيادة عدد قيم المتغيرات عن عدد أسماء المتغيرات ، وهذا عكس الحالة السابقة . ففي هذه الحالة يتجاهل الحاسب القيم الزائدة ويستمر في تنفيذ البرنامج . فإذا كانت القيمة الزائدة لا تؤدي إلى تغيير في طبيعة القيم فلن يؤثر ذلك على النتيجة .

فإذا ظهر السطر 80 كالتالي :

80 DATA 75 , 85 , 90 , 80 , 70 , 85

فستكون نتيجة حساب المعدل صحيحة .

أما إذا كان الخطأ في سطر 80 كالتالي :

80 DATA 75 , 8,5 , 90 , 80 , 70

أي أن المبرمج طبع القيمة 85 كقيمتين 8 , 5 . فهذا يؤدي إلى خطأ في نتيجة حساب المعدل ، على الرغم من أن الحاسب لم يكتشف وجوده .

٣ - عدم توافق بين تسلسل أسماء المتغيرات (صناديق) والقيم المراد تخزينها فيها .

ففي المثال السابق إذا تم إبدال تسلسل أي علامتين كما في السطر التالي :

80 DATA 85 , 75 , 90 , 80 , 70

فلن يؤثر هذا الخطأ على العملية الحسابية ، ولكن في بعض الأحيان يؤدي هذا التبديل إلى خطأ في حساب المعدل .

فإذا تم تعديل طريقة حساب المعدل بحيث أعطيت نسب مئوية مختلفة لكل واحدة من العلامات ، أصبحت عبارة 90 كالتالي :

$$90 \text{ LET } A = T1 * .2 + T2 * .15 + T3 * .25 + T4 * .10 + T5 * .30$$

أى أن نسب العلامات الخمس بالتوالى كانت :

20 % , 15 % , 25 % , 10 % , 30 %

ففى الخطأ المبين سابقاً فى عبارة 80 ستكون نتيجة حساب المعدل خاطئة ؛ بسبب تبادل نسب العلامة الأولى والثانية .

فيما سبق من هذا الفصل ، تحدثنا عن قراءة ومعالجة بيانات رقمية حسابية ، ولكن كثيراً من المشاكل التى تتم معالجتها بواسطة الحاسب تتضمن بيانات غير رقمية ولا تستعمل للعمليات الحسابية ، كاسم الموظف وعنوانه ودائرته التى يعمل بها واسم وظيفته .

وستعرض الآن لكيفية التعامل مع البيانات غير الرقمية .

تطوير برنامج باستخدام بيانات حرفية (غير رقمية) وتعريف المخرجات :

ففى المثال السابق ، لنفرض أننا أردنا تعريف اسم الدارس الذى تخصه هذه العلامات ، وهذا يتطلب منا استخدام اسم متغير غير حسابى لتخزين اسم الدارس به . وفى اختيار اسم المتغير غير الحسابى تنطبق نفس الشروط التى تنطبق على اختيار اسم المتغير الحسابى (ارجع إلى الفصل الخامس) مع إضافة إشارة الدولار \$ فى نهاية اسم المتغير .

لذا فستعدل العبارات التالية :

```
70 READ A $ , T1 , T2 , T3 , T4 , T5
```

```
80 DATA « AHMED ALI » , 75 , 85 , 90 , 80 , 70
```

لاحظ هنا أن البيانات غير الحسائية يجب وضعها بين علامتى التنصيص فى عبارة DATA للدلالة على نوعيتها غير الحسائية . ويتغير السطر 100 ليصبح كالتالى :

```
100 PRINT TAB (5); A$; TAB (15); T1; TAB (22); T2; TAB (29); T3;  
TAB (36); T4; TAB (43); T5; TAB (50); A
```

لاحظ أن المتغير A\$ تمت إضافته فى العبارة 70 فى بداية المتغيرات و تم وضع «AHMED ALI» فى بداية البيانات (80) أى أن البيانات قد رتبّت على حسب ما يرادفها من أسماء المتغيرات ونوعيتها .

ولو قمنا بتشغيل البرنامج شكل (٦ - ٢ أ) بعد إتمام التعديلات عليها ، لوجدنا أن المخرجات ستكون على الشكل التالى :

Ahmed Ali 75 85 90 80 70

دون أى تمييز لما تمثله أرقام العلامات .

فإذا أردنا تحديد الامتحانات والعلامة التى حصل عليها Ahmed Ali فى كل منها ، يمكننا عمل ذلك باستخدام تعليمة اطبع RPINT ، وهذا النوع من العمليات يطلق عليه «عمليات التعريف لحقول البيانات المخرجة عن طريق وضع عناوين لكل منها» ، ويتم ذلك قبل البدء فى طباعة أى من مخرجات البرنامج .

ففى مثالنا السابق إذا أردنا وضع عناوين لكل من حقول البيانات نستطيع عمل ذلك بإضافة العبارة التالية :

```
15 PRINT TAB (5); «NAME»;TAB (5); «TEST1»; TAB (20);  
«TEST2»;TAB (25); «TEST3»; TAB (30); «TEST4»;TAB (35); «TEST5»;  
TAB (40); «AVERAGE»
```

ولفصل العناوين عن البيانات المخرجة لتسهيل عملية قراءة وملاحظة المخرجات ، نعمل على وضع خط فاصل بإدخال العبارة التالية :

```
17 PRINT TAB (5); «.....»;«TAB (15); «.....»; TAB (20);  
«.....»;TAB (25); «.....»; TAB (30); «.....»;TAB (35); «.....»;  
TAB (40); «.....»;
```

ولنر الحل كاملاً للمثال شكل (٦ - ٢) بعد إجراء التعديلات عليه :

شكل (٦ - ٢ أ)

برنامج لحساب معدل خمس علامات لدارس واحد مع طباعة الاسم والعناوين	
10 REM	استدعاء برنامج فرعي لحساب المعدل العام و طباعة النتائج
20	GOSUB 40
30	END
40 REM	طباعة العناوين
50	PRINT TAB(5);"NAME" TAB(15);"TEST-1"; TAB(24);"TEST-2"; TAB(32);"TEST-3"; TAB(40);"TEST-4"; TAB(48);"TEST-5"; TAB(56);"AVERAGE"
60	PRINT TAB(5);"-----" TAB(15);"-----"; TAB(24);"-----" TAB(32);"-----"; TAB(40);"-----" TAB(48);"-----"; TAB(56);"-----"
70 REM	لقراءة الدرجات الخمسة و ايجاد المعدل العام و طباعته
80	READ N\$,T1,T2,T3,T4,T5
90	DATA "AHMED ALI",75,85,90,80,70
100	LET A = (T1 + T2 + T3 + T4 + T5) / 5
110	PRINT TAB(5);N\$; TAB(15);T1; TAB(24);T2; TAB(32);T3; TAB(40);T4; TAB(48);T5; TAB(56);A
120	RETURN

ولو قمنا بتشغيل البرنامج شكل (٦ - ٢ أ) لوجدنا أن المخرجات ستكون على

الشكل التالي :

NAME	TEST-1	TEST-2	TEST-3	TEST-4	TEST-5	AVERAGE
-----	-----	-----	-----	-----	-----	-----
AHMED ALI	75	85	90	80	70	80

تكلمنا في الجزء السابق من هذا الفصل عن كيفية تخزين البيانات في البرنامج المصدري عن طريق استخدام تعليمتي READ / DATA . وسنتحدث الآن عن كيفية إدخال البيانات بواسطة لوحة المفاتيح وذلك باستخدام تعليمة أدخل INPUT .

تطوير برنامج باستخدام تعليمة أدخل INPUT :

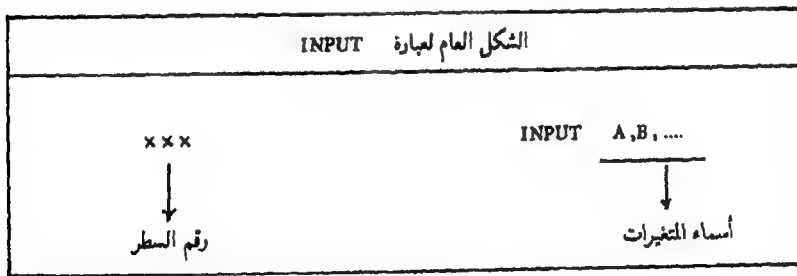
مثال (٦ - ٣) :

الهدف : إيجاد مساحة ومحيط مستطيل إذا علم طوله وعرضه .

أولاً - خطوات الحل :

- ١ - الحصول على طول وعرض المستطيل .
- ٢ - إيجاد المساحة .
- ٣ - إيجاد المحيط .
- ٤ - طباعة الطول ، العرض ، المساحة ، والمحيط .

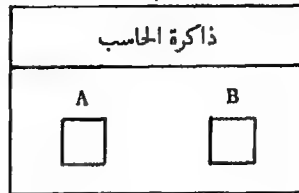
واليك الشكل العام لعبارة INPUT



إن عدد قيم البيانات المراد إدخالها يتأثر بعدد أسماء المتغيرات التي تظهر بعد كلمة INPUT .

ففى مثالنا هذا نحتاج إلى صندوقين داخل الذاكرة : واحد للطول ، وآخر للعرض .
وعليه فإن العبارة ستكون :
10 INPUT A , B

حيث يرمز A للطول و B للعرض ، أما فى الذاكرة فتكون هذه المتغيرات كالتالى :



وفى حالة تنفيذ هذه العبارة سيتوقف تنفيذ البرنامج منتظراً من المستخدم إدخال قيمتين ، ومن ثم إشعاره بالانتهاء من ذلك عن طريق ضغط مفتاح ENTFR بعد طباعة البيانات .

ثانياً وثالثاً — كما فى المثال السابق .

رابعاً : ويمثل الشكل التالى برنامجاً كاملاً وفق الأساليب الحديثة فى البرمجة .

شكل (٦ - ٣)

برنامج لحساب مساحة وعيط مستطيل إذا علم طوله وعرضه	
10	REM استدعاء برنامج فرعي لحساب المساحة والمحيط
20	GOSUB 100
39	END
100	REM برنامج فرعي للحصول على طول و عرض المستطيل
110	INPUT A,B
120	REM لحساب المساحة
130	LET C = A * B
140	REM لحساب المحيط
150	LET D = (A + B) * 2
160	REM طباعة الطول ، العرض ، المساحة و المحيط
170	PRINT A,B,C,D
180	RETURN

فلو قمنا بتشغيل هذا البرنامج فستظهر علامة استفهام على شاشة النهاية الطرفية ، مشيرة إلى أن تنفيذ البرنامج قد توقف عندما وصل إلى عبارة 110 منتظراً من المستخدم أن يدخل قيمتي كل من الطول والعرض A , B كما هو مبين فيما يلي ، ليعمل على تخزينهما (القيمتين) في الذاكرة الرئيسية .

RUN

?

وكما هو الحال في تعليمة DATA بأن يجب أن تكون البيانات مفصولة بعضها عن بعض بفاصلة .

فهنا يقوم المستخدم بواسطة لوحة المفاتيح بإدخال القيمتين مفصولتين بفاصلة . فلو تم إدخال 2 , 3 سيبقى التنفيذ متوقفاً ، إلى أن يتم إشعار البرنامج بالانتهاء من إدخال القيم بالضغط على مفتاح العودة RETURN أو ENTER حسب نوع الجهاز المستخدم (وفي كليهما ستكون الوظيفة واحدة) . وبعد الإشعار ، سيستمر تنفيذ البرنامج .

في مثالنا هنا سيتم التنفيذ كما يلي :

RUN

3 , 2

3	2	6	10
---	---	---	----

← اضغط Enter

المذكرات التوضيحية : لاحظ أن علامة الاستفهام ظهرت على الشاشة ، دون أن يسبقها أى تفسير لنوعية وترتيب وعدد البيانات . هذا يؤدي إلى غموض وبلبلة لمستخدم البرنامج .

لذا فمن المستحسن أن يسبق عبارة INPUT طباعة مذكرة توضيحية كوسيلة لتعريف حقول البيانات ، وتسهيلاً للمستخدم لتحديد أى منها يراد إدخاله ، وتسلسلها المطلوب . ويتم ذلك باتباع أحد أسلوبين :

أ — باستخدام تعليمة اطبع PRINT في سطر يسبق السطر الذى تظهر معه تعليمة أدخل INPUT . ويتم تعديل البرنامج السابق كالتالى :

```
100 PRINT      «أدخل طول المستطيل»
112 INPUT A
114 PRINT      «أدخل عرض المستطيل»
116 INUT B
```

لاحظ هنا أن المفتاح ENTFR أو RETURN سيضغط مرتين ، الأولى : بعد طباعة الطول 3 ، والثانية : بعد طباعة العرض 2 .
إذا تم تشغيل البرنامج فسيظهر التالى :

RUN			
أدخل طول المستطيل			
?	3		
أدخل عرض المستطيل			
?	2		
3	2	6	10

اضغط Enter ←

اضغط Enter ←

ب — بطباعة المذكرة التوضيحية في السطر الذى تظهر فيه تعليمة INPUT وبعدها مباشرة ، وذلك كالتالى :

```
110 INPUT      «أدخل طول المستطيل» A
116 INPUT      «أدخل عرض المستطيل» B
```

لاحظ رمز «;» الذى يسبق أول متغير يظهر فى عبارة INPUT . وإذا تم تشغيل البرنامج فسيظهر كالاتى (وذلك بعد نحو السطرين 112 , 114)

RUN			
3 ؟ أدخل طول المستطيل			
2 ؟ أدخل عرض المستطيل			
3	2	6	10

اضغط Enter ←

اضغط Enter ←

وعند تشغيل البرنامج سيتبين المغزى من استخدام تعليمة أدخل INPUT مع المذكرة التوضيحية :

- ١ - اتباع نظام المخاطبة (التعامل بطريقة مباشرة) مع الحاسب .
 - ٢ - إمكانية استخدام بيانات مختلفة فى كل مرة يتم فيها تنفيذ البرنامج دون الحاجة للتغيير فى البرنامج المصدري (تغيير البيانات الموجودة فى عبارة DATA) .
 - ٣ - عدم المزج فى نوعية وترتيب (تسلسل) البيانات .
 - ٤ - إدخال البيانات المراد التعامل معها دون زيادة أو نقصان .
- وكذلك من الممكن كتابة البرنامج بأسلوب تطلب فيه القيمتان فى سطر واحد ، وذلك كالتالى :

A , B ; «أدخل الطول والعرض بينهما فاصلة» INPUT 110

وعند تشغيل البرنامج سيحدث التالى (بعد نحو سطر 116) :

RUN			
2 , 3 ؟ أدخل الطول والعرض بينهما فاصلة			
3	2	6	10

اضغط Enter ←

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ - بين الأخطاء إن وجدت في كل من العبارات التالية :

```
10 READ A , B , C
10 INPUT A , B , C
10 READ 6 A , A  $\times$  , C
10 INPUT AB ; C ;
10 DATA 100 , 200 AB
10 DATA 500 , 600 , 101 ;
```

٣ - بين الأخطاء إن وجدت في أجزاء البرنامج التالية :

```
10 READ A , B , C
20 LET A = A + BC
30 DATA 10 , 20
10 INPUT A , B
20 LET C = A * B
30 PRINT A , BA * B
```

٤ - بين نتائج تنفيذ البرامج التالية :

```
10 READ A , B , C
20 LET D = A + B * C
30 PRINT A , B , C , D
40 DATA 10 , 5 , 2
50 END
```

```

10  READ      N$ T1 , T2
20  DATA     « ALI AHMED » , 70 , 90
30  LET       A = (T1 + T2) / 2
40  PRINT     TAB 10 , N$ , T1 , T2 , A
50  END

```

٥- باستخدام تعليمة أدخل INPUT اكتب برنامجاً يعمل على استقبال اسم الدارس وعنوانه المؤلف من : صندوق البريد ، المدينة ، والرمز البريدي ، ومن ثم طباعة هذه البيانات حسب الصيغة التالية :

الاسم :

ص . ب :

اسم المدينة :

الرمز البريدي :

ملاحظة : استخدم تعليمات الإيضاح لتحديد البيانات المطلوب إدخالها .

٦- يقوم دهان (طراش) منازل بحساب ثكلفة دهان المنزل على الأساس التالى :

الثكلفة الكلية = (المواد + الأيدى العاملة) × عامل الجودة .

ولتقدير الثكلفة الكلية يقوم الدهان بحساب مساحة الجدران بالأمتار المربعة ويطرح منها مساحة الشبايك . وتكلفة المواد هي ٧٠ ريالاً لكل ٥ أمتار مربعة . أما الأيدى العاملة فتكلف ١٢٠ ريالاً لكل ١٠ أمتار مربعة . أما عامل الجودة فيتراوح بين ٩٠٪ - ١١٠٪ وذلك حسب طبيعة الدهان فى المنزل الحالى ، وطبيعة الجدران . فإذا كانت الجدران فى وضع مقبول ولا تحتاج من التجهيز إلا إلى الجهد العادى ، فالنسبة تكون ١٠٠٪ . أما الجدران التى تتطلب تجريدها من الدهان القديم وفيها خروق ... إلخ ، فتكون النسبة ١١٠٪ ، أما الجدران التى فى وضع جيد ولا تتطلب أى تجهيز (كالمنازل الجديدة) فالنسبة هي ٩٠٪ ... وهكذا .

طور برنامجاً لحساب تكلفة المنازل إذا علمت المدخلات التالية :

مساحة الجدران ، مساحة الشبايك ، عامل الجودة .

الدوارة البسيطة والتحكم بها

مقدمة عن استعمالات الدوارة في معالجة البيانات :

في الفصلين السابقين تحدثنا عن برمجة خطوات متتالية لحل مشكلة حسابية بسيطة . وانتقلنا في العرض من إسناد قيم المدخلات إلى متغيراتها (الصناديق) ، باستخدام تعليمة LET ، ومن ثم باستعمال تعليمتي READ /DATE ، وأخيراً باستعمال تعليمة INPUT . وذلك لقراءة البيانات الموجودة داخل البرنامج المصدرى ، أو المدخلة عن طريق لوحة المفاتيح على التوالى .

وقد أوضحنا كيف أن الانتقال في عملية قراءة البيانات قد أدى إلى زيادة وضوح وفعالية البرنامج . فمع عملية الإسناد ، استدعى تغيير قيمة المتغيرات في البرنامج نفسه . أما READ / DATE فالتغيير يتم في عبارة DATA ويتم استخدام نفس المتغيرات في عبارة READ . وأخيراً في تعليمة INPUT يتم تنفيذ البرامج بطباعة RUN مرات عديدة حسب تكرار البيانات المدخلة .

وهذا التبسيط في أسلوب البرمجة يهدف إلى التدرج في عملية البرمجة ، إذ أن واقع الحال يتطلب عبارات متعددة أخرى ومعقدة ويصبح معها هذا الأسلوب عقيماً أيضاً . ويوضح المثال التالى عدم فاعلية هذا الأسلوب ، وكيفية تطويره ليتلاءم مع متطلبات التطبيقات المختلفة ، وسنستخدم نفس المثال طوال هذا الفصل لشرح التعليمات الخاصة بالتحكم بالدوارة البسيطة .

مثال (٧ - ١) :

الهدف : حساب محيط الدائرة إذا علم نصف قطرها .

خطوات الحل ومايقابلها من عبارات بيسك :

١ - الحصول على المدخلات 10 READ N

20 DATA 10

٢ - حساب المحيط = ع نق ط $3.147 * N * 4 = M$ LET M = 30

حيث نق = نصف القطر

ط = ٣,١٤٧

٣ - طباعة النتائج 40 PRINT N,M

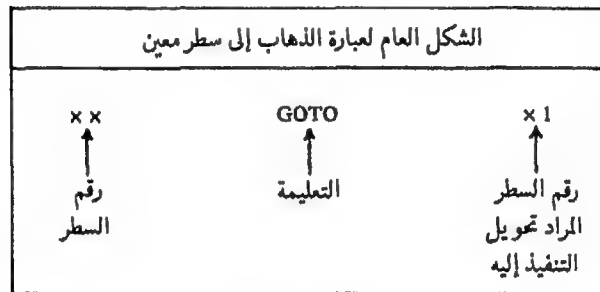
وإذا أردنا تنفيذ هذا البرنامج لحساب محيط دائرة أخرى ذات قطر مختلف ، فعلينا تغيير سطر رقم 20 وإعادة طباعة القيمة الجديدة لنصف القطر . وهذا ليس عملياً تماماً ، وخاصة إذا أردنا حساب المحيط لدوائر كثيرة . لاحظ هنا أن الخطوات لإيجاد محيط الدائرة لا تتغير ، وإنما تبقى كما هي ، فقط تتغير قيمة نصف القطر الخاص بكل دائرة . وتحدثنا في الفصل السابق عن إمكانية طباعة قيم متعددة في عبارة DATA . والفرق هنا أن هذه القيم كانت تعود إلى عنصر واحد - الطالب مثلاً . ولكن القيم في مثالنا هنا تعود إلى عناصر موصوفة - الدائرة هنا - مختلفة ، ولكن المبدأ واحد . أى أنه باستطاعتنا أن ندرج في السطر 20 جميع قيم أنصاف أقطار الدوائر المراد حساب محيطاتها . فتظهر العبارة كالتالى :

20 DATA 10 , 15 , 20 , 25 , 5 , 3 , 12 , 15 , 12

أى أن هناك ٩ دوائر يراد حساب محيطاتها ، بالإضافة إلى هذه الخاصية ، فلا بد من وجود عبارة توجه تنفيذ البرنامج إلى الخطوة الأولى مرة أخرى أو مرات ، طبقاً لعدد القيم الموجودة في عبارة DATA . وهذا ما سنبحثه في الجزء التالى .

تطوير برنامج بواسطة استخدام تعليمة الذهاب إلى رقم سطر معين :

إن تعليمة الذهاب إلى رقم سطر معين تفرض مساراً محدداً لاتجاه تنفيذ البرنامج دون أية شروط مسبقة ، لذلك يطلق عليها تعليمة التشعب غير المشروط للتحكم في مسار تنفيذ خطوات البرنامج . أى أن الحاسب عند تنفيذه هذه التعليمة ، سيحول مسار التنفيذ إلى السطر المذكور دون الأخذ بعين الاعتبار بأية شروط أو لآى من قيم المتغيرات الموجودة في البرنامج . و يوضح الشكل التالى الصورة العامة لهذه التعليمة .



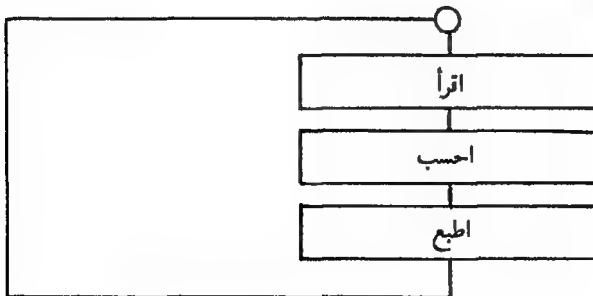
وفيما يلي خطوات تطوير برنامج للمثال السابق (٧ - ١) :

أولاً - خطوات الحل :

إضافة خطوة رقم ٤ - اذهب إلى الخطوة الأولى .

ثانياً - الهيكل الهرمى : كما في البرامج السابقة ، إلى برنامج جزئى واحد .

ثالثاً - نمط البرمجة الهيكلية :



رابعاً - البرنامج الخاص بحساب محيط الدائرة :

شكل (٧ - ١)

برنامج لحساب محيط الدائرة باستخدام عبارة GOTO	
10	REM البرنامج الرئيسي
20	REM استدعاء برنامج فرعي
30	GOSUB 100
40	END
100	REM برنامج فرعي لحساب محيط الدائرة
110	READ N
120	DATA 10, 15, 20, 25, 5, 3, 12, 45, 12
130	LET M = 4 * N * 3.14
140	PRINT N; " نصف القطر "
150	PRINT M; " محيط الدائرة "
160	GOTO 110
170	RETURN

إن البرنامج السابق يبدو أولاً ولهة أنه برنامج صحيح ، ولكن إذا أمعنت في النظر فيه ، فستجد أن إضافة عبارة الذهاب إلى السطر 160 أدى إلى تكرار تنفيذ سطور البرنامج دون أى إشارة أو تبيان لأسلوب توقيفها . فكما قلنا سابقاً ، إن عبارة GOTO تدعو إلى التشعب غير المشروط .

فإذا تم تنفيذ البرنامج ، فإنه في المرة الأولى التى تنفذ فيها السطور ، يتم أخذ قيمة 10 وهى أول قيمة فى عبارة DATA ، وتخزن فى المتغير N ، وفى المرة الثانية 15 ، وهكذا إلى نهاية القيم ، مساحة القيم السابقة التى خزنت فى N فى كل مرة حين قراءة القيمة

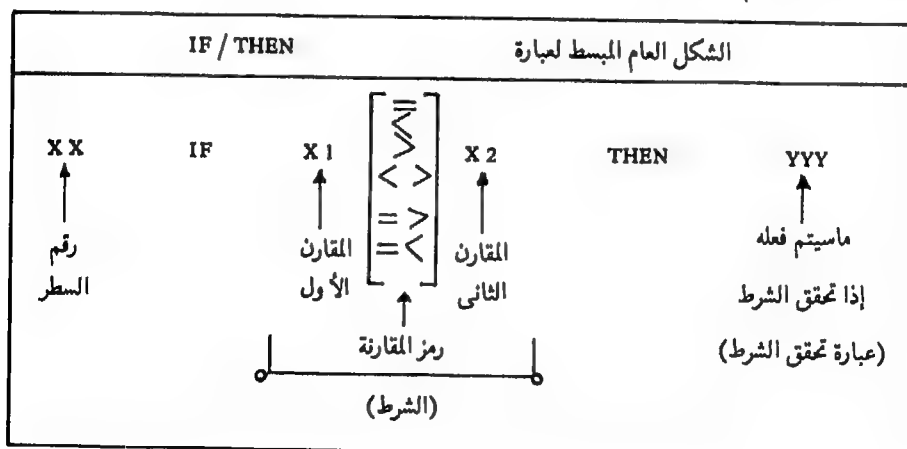
الجديدة . وبعد الانتهاء من القيمة الأخيرة ، 12 فإن البرنامج يعود إلى تنفيذ السطور مرة أخرى ، ويذهب باحثاً عن قيمة جديدة للمتغير N حين تعرضه للسطر 110 ، ولكن هذه القيم تكون قد نفذت . وحينئذ يتوقف البرنامج عن التنفيذ ، مشيراً إلى وجود خطأ ناتج عن عدم احتواء عبارة DATA على بيانات أخرى .

وبعبارة أخرى ، فإن البرنامج لا يستطيع معرفة ما إذا نفذت قيم المتغيرات تلقائياً ، وعلى المبرمج أن يتحكم بذلك عن طريق خطوات البرنامج .

وهذا يدعونا إلى التحدث عن أساليب التحكم في قراءة البيانات ؛ وذلك منعاً لتوقف البرنامج بصورة غير طبيعية عند نفاذها . ويتم ذلك عن طريق استخدام تعليمة التشعب المشروط IF / THEN .

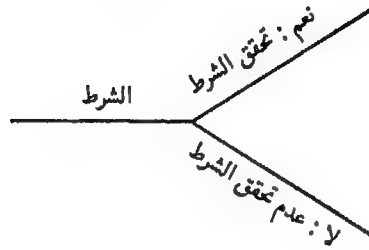
كيفية عمل تعليمة إذا / إذن IF / THEN بشكلها المبسط :

تستخدم هذه التعليمة للتحكم في مسار تنفيذ أسطر البرنامج بناء على شرط محدد ، ولها الشكل العام التالي :



ولعبارة IF/THEN جزءان رئيسيان :

١ - الشرط : إن جزء الشرط هو الذى سيحدد اتجاه مسار التنفيذ . وتكون نتيجة الشرط مرهونة بعملية المقارنة بين عنصرين من البيانات ، أطلق عليهما «المقارن الأول» و «المقارن الثانى» . وتكون هذه النتيجة ثنائية الاختيار ، أى أن التشعب عند التنفيذ يتم باختيار أحد مسارين فقط . ويوضح الشكل التالى هذا الأمر :



و يتكون جزء «الشرط» من :

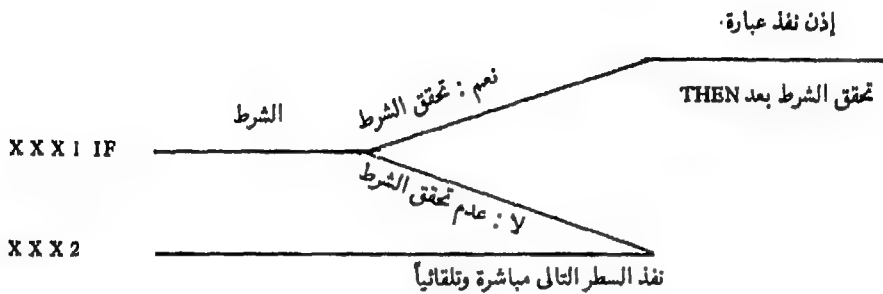
أ - المقارن الأول والمقارن الثانى : وهذه تمثل البيانات سواء كان ذلك قيمة أو بما تحتويها أسماء المتغيرات من قيم . أى أن X_1 و X_2 قد تأخذان أرقاماً ثابتة مثل ٦ ، ١٠ ، ٥ ، ٢ ، إلخ ، أو أسماء لمتغيرات (صناديق) مثل M_1, K, N ... إلخ . ويشترط أن تخزن قيمة محددة فى المتغير المستخدم فى عبارة IF/THEN قبل تنفيذها .

ب - رمز المقارنة : ويستخدم هذا الرمز للمقارنة بين المقارن الأول والثانى ، ويمكن أن يأخذ أحد الرموز التالية :

الرموز المستخدمة فى المقارنة بعبارة IF / THEN	
يساوى	=
أصغر من	<
أكبر من	>
يساوى أو أكبر من	>=
يساوى أو أصغر من	<=
لا يساوى	< >

٢ - عبارة تحقق الشرط : وهى العبارة التى سيتم تنفيذها فى حالة ما إذا كانت نتيجة المقارنة قد تحققت . وإحدى العبارات هذه قد تكون عبارة GOTO X X X أو غيرها . وسنكتفى هنا بالتحدث عن كيفية تنفيذ عبارة IF/THEN إذا أدرجت معها تعليمة GOTO . وفى حالة عدم تحقق الشرط فإن البرنامج سينفذ التعليمة الموجودة فى السطر الذى يلى مباشرة عبارة IF/THEN

و يوضح الشكل التالى هذا التسلسل :



وسنوضح الآن كيفية استخدام عبارة IF / THEN لإيقاف تنفيذ البرنامج بشكل طبيعى ، أى بعد نفاذ البيانات .

تطوير برنامج باستخدام IF / THEN لإيقاف التنفيذ طبيعياً :

إن الوصول إلى معرفة اللحظة التى تتم فيها معالجة آخر قيمة للبيانات ، يمكن تحديده بإحدى وسيلتين : باستخدام قيمة وهمية ، أو بالمعرفة المسبقة لعدد البيانات .
أ - وضع قيمة وهمية فى نهاية القيم التى يراد معالجتها ، بحيث يوقف تنفيذ البرنامج عند قراءة هذه القيمة الوهمية النهائية ، وقبل تنفيذ الأسطر من البرنامج والتى تؤدي إلى معالجتها . وهذا التوقيف واضح السبب ، إذ أن هذه القيمة الوهمية ، بالتعريف ، هى فقط لتؤدي مهمة معينة وليس لهدف معالجتها .

إن الخطوات المعطاة للمثل السابق قد أعيدت هنا لتبيان كيفية تعديلها لتوافق هذا الأسلوب ، وسنفرق مع البرامج نمط البرمجة التركيبية لتتسع التغييرات والاختلافات بين الأساليب المختلفة لإيقاف تنفيذ البرنامج .

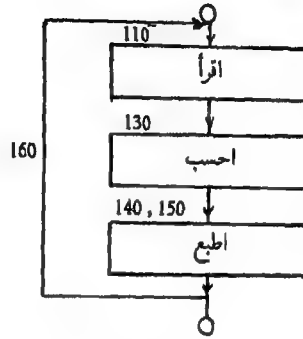
البرنامج

```

110 READ N
120 DATA 10 , 15 , 20 , 25 ,
      5,3,12,15,12
130 LET M = 4 * N * 3 . 14
140 PRINT «نصف القطر» ; N
150 PRINT «مساحة الدائرة» ; M
160 GO TO 110
120 RETURN

```

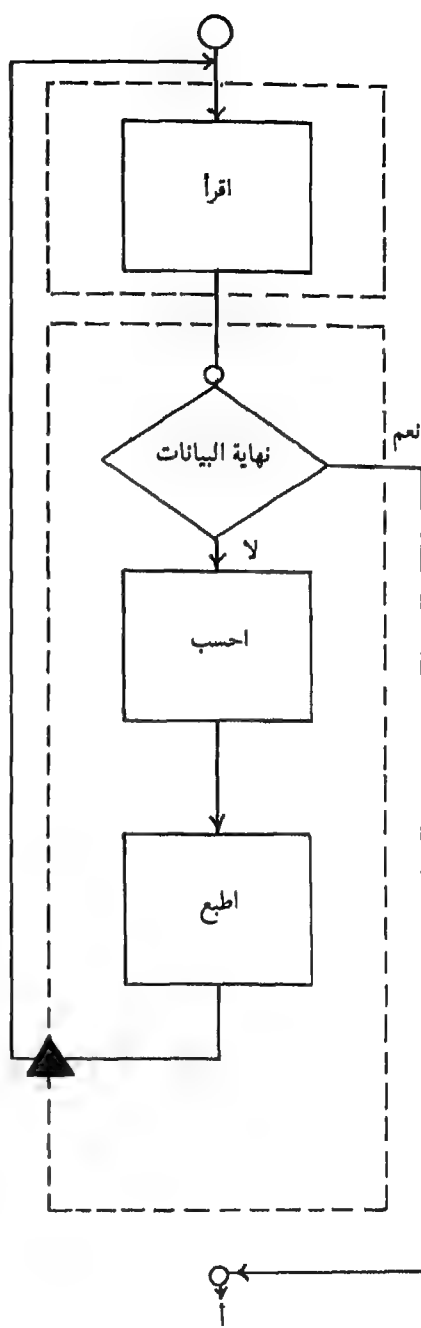
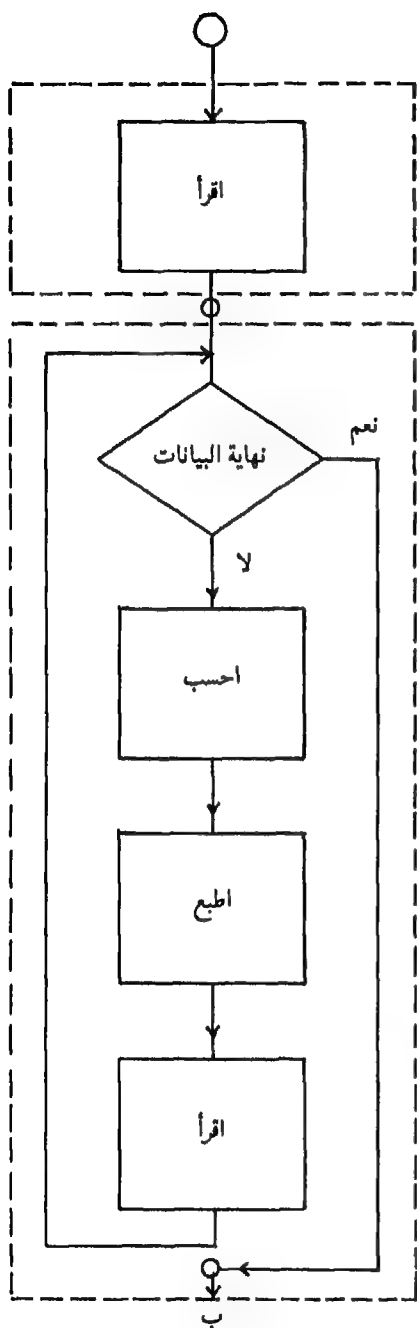
نمط البرمجة التركيبية



فتلاحظ بوضوح كيف أن البرنامج يدخل في دوارة لانتهائية ، وأن الدائرة الصغيرة في نهاية الشكل لن يحدث تشعب لها . وإذا تمت إضافة قيمة وهمية ، — ١ مثلاً ، في نهاية البيانات في السطر 20 ، يصبح كالتالي :

```
120 DATA 10 , 15 , 20 , 25 , 12 , 15 , 12 , 1 =
```

فمن الممكن استخدامها في عبارة IF/THEN بغرض توقيف البرنامج بعد قراءتها مباشرة . وقبل صياغة عبارة IF/THEN لابد من معرفة موقعها منطقياً في تسلسل الخطوات . فالقراءة تتم في الخطوة الأولى والمعالجة في الثانية ، إذن لابد من ظهور عبارة IF/THEN بينهما . ويصبح الشكل كالتالي ، شكل أ ، وهذا الشكل لايناسب مبادئ البرمجة التركيبية تماماً ؛ لأن تحديد بدايات ونهايات الأجزاء ليس واضحاً ، ومن الممكن تطوير أسلوب أفضل كما في الشكل ب ، وذلك بزيادة خطوة القراءة مرة أخرى في النهاية .



لاحظ الخطوط المتقطعة التي تحدد الأجزاء ، تجد أنه في الشكل «أ» يقاطع خط التشعب العائد إلى أعلى عند المثلث المظلل في حين تم تقسيم الشكل «ب» إلى جزأين متتابعين مستقلين .

ولتعديل البرنامج السابق المذكور في شكل (٧ — ١) علينا إضافة عبارتي IF/THEN و READ الثانية . ولتحديد رقمي السطرين ، نعود إلى الشكل ب الذي يوضح مسار التشعب حسب أساليب البرمجة الهيكلية . ففي الخطوط الموصلة بين الرموز ، فيما عدا تلك التي توصل بين المربعات المتتالية ، يتم استخدام عبارة GOTO للدلالة على التشعب إلى سطر سابق أو إلى سطر لاحق .

بناء على ذلك فإن الخطوة التي تتضمن عبارة القراءة READ الثانية تتبع خطوة الطبع ، ومن ثم تليها عبارة الذهاب إلى الخطوة التي تقوم بعملية الحساب . أما عبارة IF/THEN فهي تلي عبارة READ الأولى . وعلى وجه التحديد يضاف السطران التاليان للبرنامج :

```
120 IF N = — 1 THEN GOTO 170
```

```
155 READ N
```

واليك الآن البرنامج الكامل إلى جانب شكل البرمجة التركيبية للمقارنة بينهما ، بالإضافة إلى تتبع عدد القراءات وما يصاحبها من تغيير في القيمة المخزنة في N .

شكل (٧-٢)

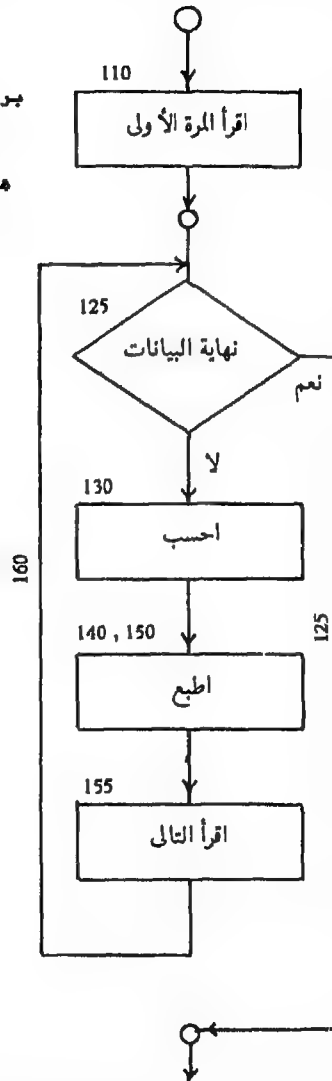
برنامج لحساب محيط الدائرة باستخدام تعليمة إذا/إذن لإيقاف البرنامج
بواسطة القيمة الوهمية ورسم البرعة التركيبية

```

10 REM البرنامج الرئيسي
20 REM استدعاء برنامج فرعي
30 GOSUB 100
40 END

100 REM برنامج فرعي لحساب محيط الدائرة
105 REM أول قراءة للقيمة
110 READ N
120 DATA 10,15,20,25,5,3,12,45,12,-1
123 REM هل القيمة مؤشر نهاية البيانات
125 IF N = -1 THEN 170
130 LET M = 4 * N * 3.14
140 PRINT N; " نصف القطر "
150 PRINT M; " محيط الدائرة "
152 REM n التالي للقيم
155 READ N
160 GOTO 125
170 RETURN

```



مايجرى للقيمة المخزنة داخل الصندوق N	
عدد القراءة	قيمة N
الأولى	10
الثانية	15
الثالثة	20
الرابعة	25
.....
التاسعة	12
العاشر (التوقف عن المعالجة)	-1

لاحظ أن قراءة القيمة الجديدة للمتغير N تمحو القيمة السابقة لها ، كذلك فإن آخر قيمة مخزنة في N هي ١ - ، أى القيمة الوهمية .

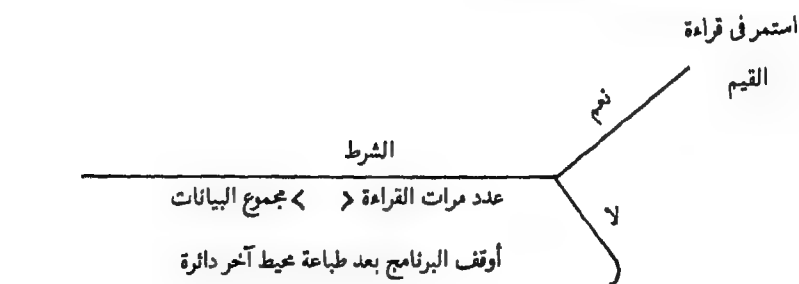
لقد شرحنا في القسم السابق استخدام القيمة الوهمية في نهاية قيم البيانات ، وستعرض الآن للأسلوب الثانى .

ب - معرفة عدد البيانات المدخلة : لنفترض أننا على علم بأن عدد الدوائر المراد حساب مساحتها هي ٩ ، فبالإمكان التدقيق بعد كل قراءة لمعرفة ما إذا تم قراءة جميع البيانات . وهذا يتطلب استخدام اسم متغير - صندوق - لمتابعة ومراقبة عدد مرات

القراءة . وإليك توضيحاً للخطوات المنطقية المتبعة في تحديد نقطة نهاية البيانات :

أوقف التنفيذ	استمر في القراءة	نتيجة الشرط	الشروط	
			عدد القيم المقروءة	رمز المقارنة لا يساوي
	X	نعم	٩	< >
	X	نعم	٩	< >
	X	نعم	٩	< >
	.	.	.	
	.	.	.	
	.	.	.	
	.	.	.	
	X	نعم	٩	< >
	X	نعم	٩	< >
X		لا	٩	< >

و يزيد في التوضيح الشكل التالي :



أما خطوات الحل التفصيلية فتكون كالتالى :

- ١ - اقرأ قيمة نصف القطر .
 - ٢ - أضف «١» إلى قيمة العداد السابقة .
 - ٣ - احسب المحيط .
 - ٤ - اطبع المحيط .
 - ٥ - إذا كانت قيمة العداد = ٩ إذن توقف وإلا فاذهب إلى الخطوة الأولى .
- وتتطلب الخطوة الثانية استخدام عبارة لم يتم شرحها بعد ، وهى كيفية استخدام تعليمة الإسناد LET مع المتغير العداد COUNTER ، وذلك لمتابعة عدد مرات تنفيذ سطر أو مجموعة أسطر . ويبين الرسم التالى شكل عبارة LET مع المتغير العداد .

استخدام تعليمة LET مع عداد COUNTER				
XX	LET	X 1	= X 1	+ 1
↑ رقم السطر		↑ القيمة الجديدة	↑ القيمة السابقة	↑ الزيادة فى القيمة
		نفس المتغير العداد COUNTER		

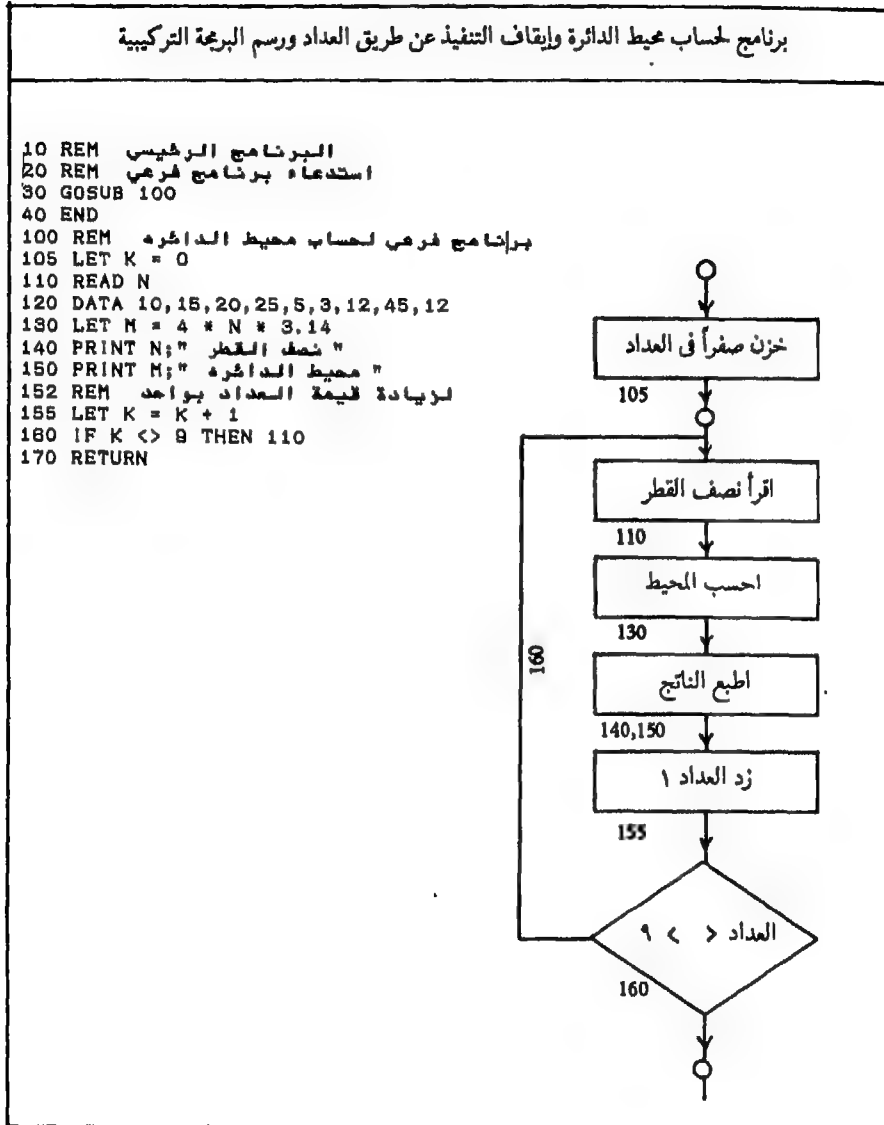
واستخدام إشارة «(=)» هنا ليس بمضمونها الجبرى ، وإلا كان ١ = صفراً ، بل هو بمعنى البرمجة . أى أن البرنامج يأخذ القيمة السابقة (الحالية) المخزنة فى العداد - الصندوق - ويضيف واحداً لها ، ومن ثم يعود إلى تخزين القيمة الجديدة فى نفس الصندوق . وإليك توضيحاً لما يجرى داخل الذاكرة ، على فرض أن القيمة السابقة = صفراً .

لحاسب

القيمة الحالية : قبل تنفيذ تعليمة LET
X 1
0

و يوضح الشكل التالى البرنامج المعدل لحساب محيط الدائرة، حسب هذا الأسلوب
 فى إيقاف تنفيذ البرنامج، مع شكل البرمجة التركيبية المناسب .

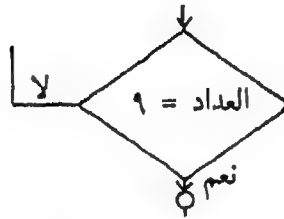
شكل (٧ - ٣)



لاحظ استخدام إشارة > بدلا من = لتسهيل عملية كتابة البرنامج . ولوأردنا استخدام = لكان شكل البرمجة التركيبية كما في السابق ، سوى تبديل كلمتى «نعم» و «لا» إحداهما مكان الأخرى . أما البرنامج فسيتم فيه الأسطر الخاصة بإنهاء تنفيذ البرنامج على الشكل التالى ، و يليه الجزء المعدل من رسم البرمجة التركيبية :

160 IF K = 9 GOTO 170

165 GOTO 110



لذلك كان الاختلاف في رسم البرمجة التركيبية لايتغير جوهرياً ، في حين ساعد استخدام > بدلاً من = في السطر 165 في التخلص من ضرورة استخدام واحدة من عبارات التشعب GOTO .

لاحظ وجود السطر 105 الذى يفوم بوضع قيمة أولية «صفر» في العداد «K» ، وهذا ينطبق على جميع المتغيرات التى تستعمل كعدادات ، ويجب أن يتم ذلك قبل الوصول إلى مرحلة تنفيذ السطر الذى تظهر فيه عبارة LET مع العداد .

ويختلف شكل ٢ عن شكل ١ بالأسطر التالية فقط :

105 , 155 , 160

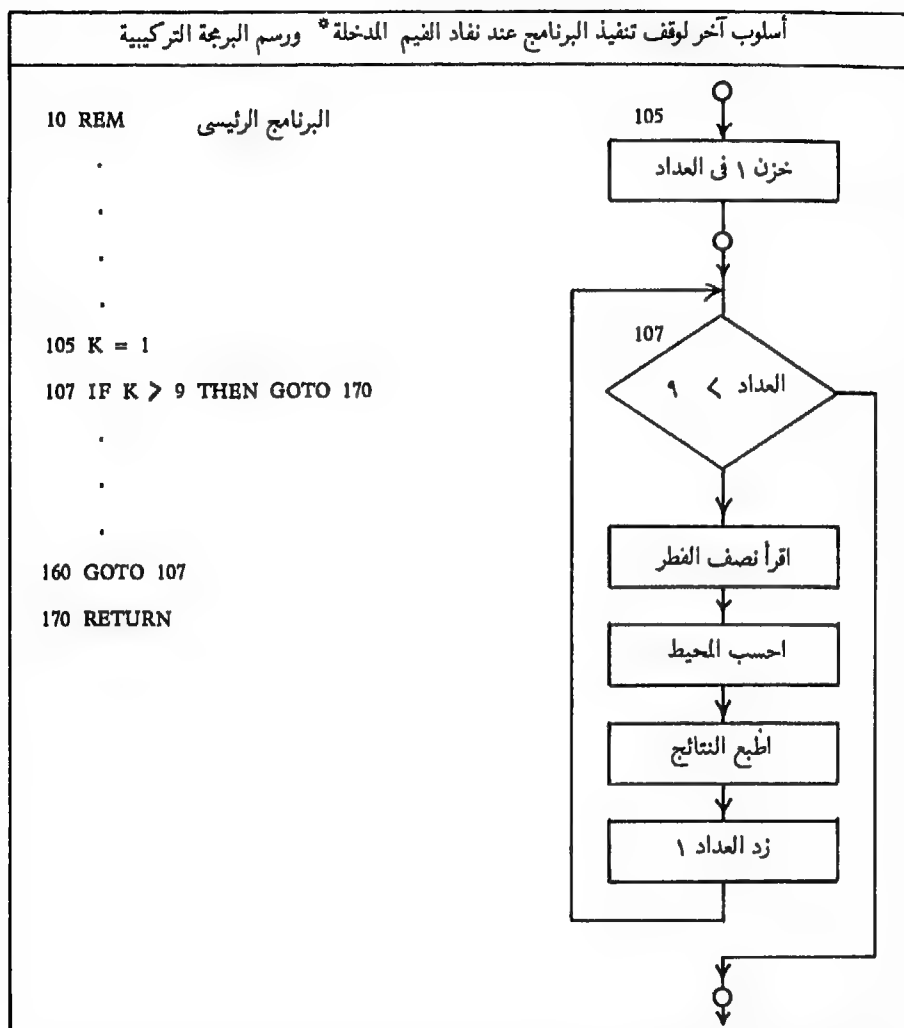
ومن الممكن متابعة تسلسل قراءات القيم بطريقة أخرى ، وإن كانت في جوهرها مشابهة للطريقة السابقة . والاختلاف هو في توقيت وكيفية تدقيق انتهاء قيم البيانات .

فإذا أردنا تدقيق ما إذا كان العداد يزيد على ٩ ، فإن ذلك يجب أن يتم في بداية البرنامج وقبل قراءة القيمة التالية لنصف القطر . وذلك لتجنب الخطأ الناتج عن تنفيذ

عبارة الفراءة للمرة العاشرة دون وجود بيانات . ولما كان التدقيق يتم قبل القراءة ، فإن القيمة الأولية للعداد يجب أن تكون ١ وذلك ليشير إلى القراءة الأولى .

واليك البرنامج حسب هذه الطريقة الجديدة مع رسم البرمجة التركيبية المناسب لها .

شكل (٧ - ٣)



«ذكرت هنا الأسطر المختلفة فقط عن البرنامج السابق

وقد اضطررنا إلى استخدام عبارتين للشعب غير المشروط GOTO ، وإن كان بالإمكان العودة إلى البرنامج الرئيسى من سطر 107 إذا ظهر على الشكل التالى :

```
107 IF K > 9 THEN RETURN
```

ولكن أساليب البرمجة التركيبية تتطلب الشعب إلى نهاية الجزء . وفى حالة تكرار أماكن العودة إلى البرنامج الفرعى من أسطر مختلفة ، فمن الأفضل الإشارة إلى تعليمة RETURN مرة واحدة ، وذلك منعاً للإرباك ولسهولة تتبع تنفيذ البرنامج . وسيتم الشرح التفصيلى للشكل المذكور سابقاً لتعليمة IF/THEN ، والذي يسمح بإيراد تعليمات أخرى غير الذهاب إلى سطر GOTO بعد تعليمة THEN ، فى الفصل التالى .

تمارين

١ — اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ — بين الأخطاء إن وجدت في اختيار أسماء المتغيرات غير العددية التالية :

A	\$	A5\$	B%
B . \$	P£	7	«R 5»

٣ — بين الأخطاء إن وجدت في القيم غير العددية التالية :

«THE NAME IS»	«HIS AGE IS 28
«P.O . Box» (404)	«THE STUDENT'S BOOK»

٤ — اختر أسماء المتغيرات الملائمة للبيانات التالية :

75.2	777 - 7777	P.O. Box
ALI	70 KPH	60 METERS

٥ — بين الأخطاء إن وجدت في كل من العبارات التالية :

10	INPUT	'ENTER NAME , AGE' , N , A	أ —
10	READ	N\$, P\$, A , N	ب —
10	PRINT	N, P , R ; S	ج —
10	<u>GOTO</u> END		د —
10	IF A 5	THEN 40	هـ —

٦- بين عدد المرات التي يتم فيها تنفيذ البرامج التالية :

10 LET I = I

أ -

20 LET I = I + I

30 LET PRINT I

40 IF I = 5 THEN 20

50 END

10 LET K = 0

ب -

20 READ A , B

30 PRINT A , B

40 IF A = 4 THEN 70

50 DATA 10 , 5 , 20 , 4 , 4 , 8 , 9 , 2

60 END

٧- بين نتائج تنفيذ البرامج التالية :

10 READ A , B

أ -

20 IF A < 2 THEN 60

30 PRINT A + B , A + B

40 DATA 5 , 4 , 2 , 1 , 3 , 1 , 5 , 2 , 1 , 4 , 2 , 6 , 7 , 3

50 COTO 10

60 END

10 READ A , B , C

ب -

20 IF A\$ = 'AA' THEN 70


```

30 PRINT A$, B, C
40 DATA ALT, 70, 80, AHMED, 70, 60, AAN, 60, 50
50 DATA SALIM, 80, 70, AA, 60, 60, OMAR, 80, 70
60 COTO 10
70 END

```

٨ — باستخدام تعليمة IF/THEN اكتب برنامجاً يعمل على طباعة الأرقام من عشرة إلى عشرين .

٩ — يتم تقويم الدارس في البرامج الإعدادية في معهد الإدارة العامة على الأسس التالية :

- ألا يقل عدد التقويمات خلال الفصل عن أربعة ، بما فيها التقويم النهائي .
- ٣٠% للتقويم النهائي .
- ألا يزيد وزن التقويم الواحد على ٢٠% .

وفي نهاية الفصل التدريبي يتم تحضير جدول تظهر فيه العلامة لكل تقويم من الوزن الخاص بها . والعلامة من ٧٠ لأعمال الفصل ، والعلامة من ٣٠ للتقويم النهائي ، والعلامة النهائية من ١٠٠ لتقويم الفصل ، وأخيراً التقويم للفصل من ٤ (أى يطرح ٥٠ من العلامة النهائية ويقسم الناتج على ١٠ ، فإذا كانت العلامة ٦٠ يكون التقويم ١ ول ٧٠ ، ٢ ول ٨٠ ، ٣ ول ٩٠ ، ٤ وهكذا أيضا للعلامات الواقعة فيما بينها) .

طور برنامجاً لطباعة الجدول ، إذا أراد المدرس إدخال العلامات من ١٠٠ وأوزانها المرادفة للتقويم الفصلية وعلامة التقويم النهائي . مع العلم بأن القوانين لا تسمح بالكسور العشرية في العلامات . ماعدا التقويم النهائي من ٤ .

تعليمات التشعب

مقدمة عن ضرورة التشعب لمعالجة البيانات :

تحدثنا في الفصل السابق عن التشعب الناتج عن انتهاء البيانات المدخلة ، والذي يستخدم تعليمة التشعب المشروط بشكلها المبسط IF/THEN مقرونة بتعليمة التشعب غير المشروط GOTO لتجاهل (القفز عن) تنفيذ مجموعة من التعليمات ، وكان التدقيق يتم إما عن طريق قيمة وهمية تظهر في آخر البيانات أو عن طريق المعرفة المسبقة بعدد البيانات المدخلة .

ولكن كثيراً من التطبيقات العملية في المجالين التجارى والحكومى ، تتطلب تفرعات مشروطة وغير مشروطة متعددة بخلاف ما تقدم . ومن أنواع هذه التفرعات : التفرع بين وظائف الأجزاء المتعددة ، والتفرع داخل الجزء الواحد .

١ - التفرع من الأجزاء : أوضحنا في الفصل الثالث أمثلة من هياكل هرمية واقعية ، ويتطلب تنفيذ البرنامج المطور لأى تركيب هرمى الانتقال من وظيفة إلى أخرى ، أى من جزء إلى آخر . وبشكل رئيسى التفرع العمودى من الجزء المتحكم إلى الأجزاء المنفذة التابعة له ، وخاصة إذا كان التشعب يستدعى اختياراً واحداً أو أكثر من الأجزاء بناء على المقارنة بين قيم أحد المتغيرات .

لنفرض أن المتغير TYP يحدد أياً من الأجزاء التنفيذية الثلاثة التالية ، إذن يجب تنفيذ 1 X , 2 X , 3 X . فإذا كانت قيمة TYP = 1 يتم تنفيذ الجزء 1 X ، وإذا كانت قيمة TYP = 2 يتم تنفيذ الجزء 2 X ، وإذا كانت قيمة TYP = 3 يتم تنفيذ الجزء 3 X .

و يقوم البرنامج الرئيسى بتوجيه التنفيذ حسب القيمة المخزنة فى TYP فى أى مرحلة من مراحل تنفيذ البرنامج .

٢ - التفرع داخل الأجزاء التنفيذية : قد يتطلب الأمر التفرع أثناء تنفيذ الجزء لوظيفته المحددة . فالجزء الذى يحسب الراتب الإجمالى يكون التفرع بناء على عدد ساعات العمل ، وذلك إذا اقتضى الأمر حساب خارج الدوام . والجزء الخاص بحساب بدلات النقل يكون التفرع فيه حسب مرتبة الموظف ... وهكذا . وفى هذه الحالات يتم التفرع بناء على تدقيق قيم متغيرات مدخلة ، وغالباً ما تتم هناك معالجات حسابية ، وطباعة مخرجات ، وقراءة مدخلات مختلفة باختلاف حالة التفرع .

وهناك حالات أخرى تتطلب التفرع ، وهى حالات التصنيف والفرز والتبويب والتلخيص للبيانات المدخلة . ومن أمثلة هذه الحالات : حساب إجمالى مبيعات المناطق الثلاث الشرقية والوسطى والغربية لشركة سيارات ، تصنيف الدرجات إلى A , B , C , D , F أى إلى امتياز ، جيد جداً ، جيد ، مقبول ، وراسب على التوالى ، تبويب الأعمار إلى فئات متتالية أقل من ١٦ ، ١٦ - ٢٠ ، ٢٠ - ٢١ ، ٢١ - ٢٥ ، ٢٥ - ٢٦ ، ٢٦ - ٣٠ ، ٣١ - ٣٥ ، ٣٦ - ٤٠ ، ٤١ - ٤٥ ، وأكبر من ٤٥ .

وكذلك فإن استخلاص بعض المعلومات من ملف البيانات المدخلة وفق شروط محددة ، يتطلب تشعباً متعددأ . ومن الأمثلة على ذلك تحضير قائمة بأسماء الموظفين التابعين للدائرة المالية أو الدائرة الإدارية ، ورواتبهم تزيد على ٨٠٠٠ ريال شهرياً ، أو معرفة أسماء العمال الذين يعملون فى إنتاج القطعة الأولى أو الثانية ، ولهم سنوات خبرة تزيد على خمس سنوات ، أو توزيع المهام حسب المستوى الأكاديمى أو عدد سنوات الخبرة والعمر... وهكذا .

وكلا النوعين المذكورين سابقاً تتطلب معالجتهما استخدام تعليمة التشعب المشروط IF/THEN المتعددة ، وتعليمات أخرى سيتم تقديمها في هذا الفصل .

وسنبحث استخدام هذه التعليمات في حالات التشعب داخل الجزء المنفذ أولاً ، ومن ثم التشعب من الجزء المتحكم إلى الأجزاء المنفذة . والنوع الأول يمثل التشعب الثنائي ، ويمثل النوع الثاني التشعب المتعدد .

التشعب الثنائي

تطوير برنامج باستخدام تعليمة IF / THEN :

مثال (٨ - ١) :

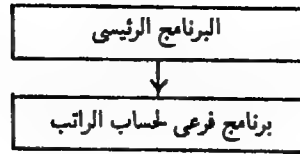
الهدف : إيجاد الراتب الإجمالي لكل موظف في مؤسسة معينة .

أولاً - خطوات الحل :

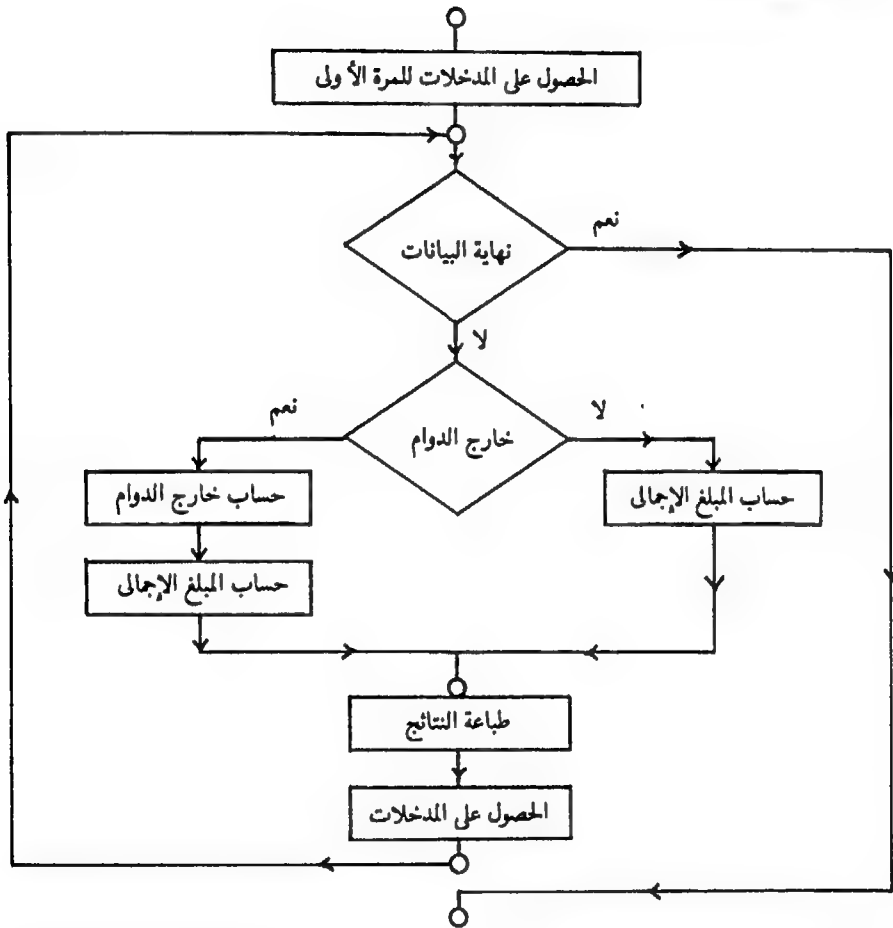
- ١ - الحصول على اسم الموظف ، عدد ساعات العمل الأسبوعية وأجرة الساعة .
- ٢ - إيجاد المبلغ المستحق لخارج الدوام إن وجد ، والمبلغ الإجمالي .
- ٣ - طباعة اسم الموظف ، عدد ساعات العمل ، الراتب الأساسي ، المبلغ المستحق من خارج الدوام والمبلغ الإجمالي .

وسيتم تطوير برنامج لتحقيق هذا الهدف يحتوي على برنامج فرعى واحد يستخدم فيه تعليمة IF/THEN مقرونة بتعليمة GOTO كما تم شرحه في الفصل السابق .

ثانياً - الهيكل الهرمي :



ثالثاً - غط البرمجة التركيبية :



لاحظ التشعب عند تدقيق خارج الدوام ، ومن ثم الاكتفاء بطباعة النتائج والحصول على المدخلات للمرات المتتالية .

رابعاً - ويحوى الشكل التالى البرنامج كاملاً حسب الخطوات المبينة سابقاً .

شكل (٨ - ١)

برنامج لحساب الراتب الإجمالى لموظفى مؤسسة معينة باستخدام عبارة IF/THEN واحدة

```

10 REM لا ستدعاء البرنامج الفرعى
20 GOSUB 50
30 END
40 REM البيانات
50 DATA "ALI AHMED",35,20
60 DATA "SALIM ALI",42,25
70 DATA "MOHAMED OMER",60,19
80 DATA "AHMED HASAN",52,28
90 DATA "dummy",0,0
100 REM لطباعة المناوبين
110 PRINT "      NAME      HOURS WORKED      RATE      BASE SALARY  OVERTIME  N
ET PAY"
120 PRINT "===== "
130 REM لقراءة اسم الموظف، وعداد ساعات العمل، اجرة الساعة
140 READ N#,H,P
150 REM لا ستفسار من نهاية البيانات
160 IF N# = "dummy" THEN 350
170 REM لا ستفسار من وجود خارج دوام
180 IF H <= 40 THEN 250
190 REM ايجاد عدد ساعات الدوام والمبلغ المستحق لها
200 LET O = ((H - 40) * 1.5) * P
210 REM لايجاد المبلغ الا جمالى
220 LET N = (40 * P) + O
230 GOTO 280
240 REM لايجاد راتب الشخص الذى لم يعمل خارج دوام
250 LET N = H * P
260 LET O = 0
270 REM لطباعة اسم الموظف، وعداد ساعات العمل، اجرة الساعة، راتب اجمالى
280 REM لايجاد الراتب الا ساسي
290 LET B = N - O
300 PRINT TAB(2);N#; TAB(20);H; TAB(37);P; TAB(48);B; TAB(58);O;TAB(87);N
310 PRINT "-----"
320 REM لقراءة اسم الموظف، وعداد ساعات العمل، اجرة الساعة
330 READ N#,H,P
340 GOTO 180
350 RETURN

```

وعند تنفيذ البرنامج ستظهر النتائج كالتالى :

NAME	HOURS WORKED	RATE	BASE SALARY	OVERTIME	NET PAY
ALI AHMED	35	20	700	0	700
SALIM ALI	42	25	1000	75	1075
MOHAMED OMER	60	19	760	570	1330
AHMED HASAN	52		1040	468	1508

في البرنامج السابق نلاحظ وجود أربع عبارات تستخدم تعليمة اذهب إلى GOTO ، وهذا مخالف إلى حد ما لقواعد البرمجة التركيبية .

لذا يمكن تعديله وكتابته بطريقة أفضل باستخدام شكل آخر لعبارة :

IF / THEN

فلايجاد :

١ — راتب الموظف الذى لم يشتغل خارج دوام، يمكننا عمل ذلك باستخدام العبارة التالية :

150 IF H <= 40 THEN N = H * P

لتحل مكان العبارات السابقة (150 , 220 , 230) .

٢ — راتب الموظف الذى يوجد له خارج دوام، تستخدم العبارة التالية :

170 IF H > 40 THEN O = (H - 40) * (P * 1.5)

ومن ثم نجد الراتب الإجمالى للموظف الذى يوجد له خارج دوام بإضافة خارج الدوام إلى الراتب الأساسى :

190 IF H > 40 THEN N = O + (P * 40)

وبإجراء هذه التعديلات تصبح الصيغة النهائية للبرنامج كالتالى :

شكل (٨ - ١)

برنامج لحساب الراتب الإجمالي لموظفي مؤسسة معينة باستخدام عبارات IF / THEN متعددة

```

10 REM لا ستدعاء البرنامج الفرعي
20 GOSUB 50
30 END
50 REM لطباعة العناوين
60 PRINT "      NAME      HOURS WORKED      RATE      BASE SALARY      OVERTIME      NE
T PAY"
70 PRINT "===== "
80 REM لقراءة اسم الموظف ،عدد ساعات العمل ،اجرة الساعه
90 READ N#,H,P
100 REM لا ستفسار من نهاية البيانات
110 IF N# ="dummy" THEN 350
115 LET O = 0
120 REM لا ستفسار من وجود خارج دوام
130 IF H <= 40 THEN N = H * P
140 REM ايجاد عدد ساعات الدوام والمبلغ المستحق لها
150 IF H > 40 THEN O = (H - 40) * ( 1.5 * P)
160 REM لا إيجاد المبلغ الا جمالي
170 IF H > 40 THEN N = (40 * P) + O
220 REM لطباعة اسم الموظف ،عدد ساعات العمل ،اجرة الساعه بخارج دوام ،راتب اجمالي
230 REM لا إيجاد الراتب الا ساسي
240 LET B = N - O
250 PRINT TAB(2);N#; TAB(20);H; TAB(37);P; TAB(48);B; TAB(58);O;TAB(67);N
260 PRINT "-----"
270 REM لقراءة اسم الموظف ،عدد ساعات العمل ،اجرة الساعه
280 READ N#,H,P
290 GOTO 110
300 DATA "ALI AHMED",80,23
310 DATA "SALIM ALI",42,25
320 DATA "MOHAMED OMER",36,19
330 DATA "AHMED HASAN",52,18
340 DATA "dummy",0,0
350 RETURN

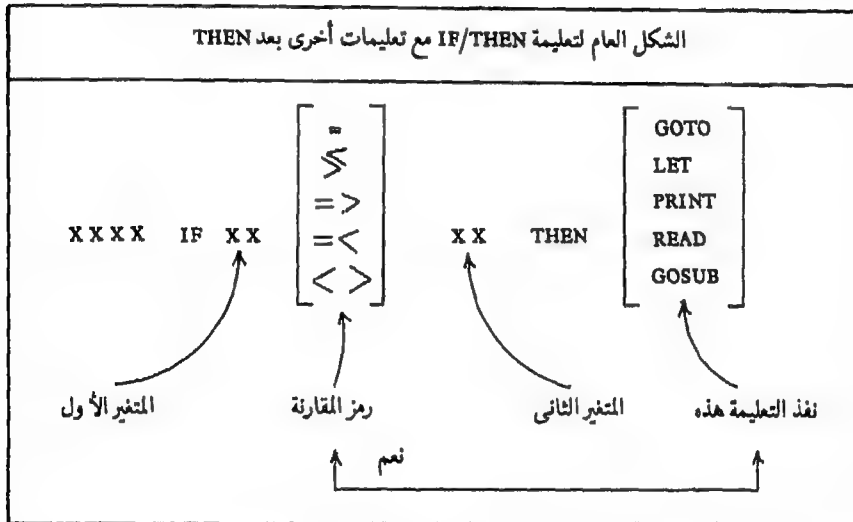
```

وفي حال تنفيذ البرنامج ، تظهر النتائج كالتالي :

NAME	HOURS WORKED	RATE	BASE SALARY	OVERTIME	NET PAY
=====	=====	=====	=====	=====	=====
ALI AHMED	80	23	920	690	1610
SALIM ALI	42	25	1000	75	1075
MOHAMED OMER	36	19	684	0	684
AHMED HASAN	52	18	720	324	1044
-----	-----	-----	-----	-----	-----

لاحظ في كل من العبارتين (150 , 170) أنه قد تم إتباع تعليمة IF/THEN بعملية حسابية ، بدلاً من إتباعها برقم العبارة التي تجرى تلك العملية الحسابية ، وبذلك تم الاستغناء عن العديد من تعليمات GOTO ، وهذا مايتوافق مع أساليب البرمجة التركيبية .

ويعكس الرسم التالى الشكل العام لتعليمة IF/THEN عند استخدام تعليمات أخرى في نفس العبارة .



أى أن البرنامج ينفذ التعليمة الظاهرة عقب كلمة THEN في حالة كون المقارنة صحيحة . أما في حالة كون المقارنة غير صحيحة ، فيتم تنفيذ السطر التالى تلقائياً .

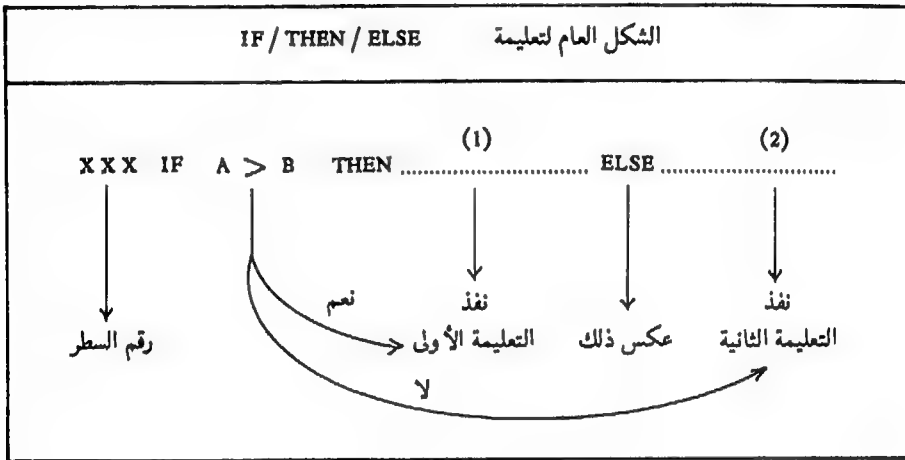
تم تعديل البرنامج السابق ، شكل (٨ - ١ أ) بتقليل عدد تعليمات اذهب إلى GOTO ليتناسب مع قواعد البرمجة التركيبية وذلك بإجراء العملية اللازم القيام بها عند تحقيق الشرط ، ولذا زاد عدد المرات التى استخدمت بها تعليمة إذا / إذن IF/THEN ، حيث قمنا بالاستفسار عن كل حالة منفردة دون تغيير منطق البرنامج .

تطوير برنامج باستخدام تعليمة IF/THEN/ELSE :

وسنتعرف الآن على كيفية التقليل من عدد تعليمات إذا/إذن وذلك باستخدام تعليمة إذا / إذن / عكس ذلك

IF / THEN / ELSE

كما هو موضح في الشكل التالي :



هنا يتم تنفيذ التعليمة الأولى (١) في حالة تحقيق الشرط ، وهو أن تكون قيمة المتغير A أكبر من قيمة المتغير B ، ولكن إذا لم تكن قيمة المتغير A أكبر من قيمة المتغير B يتم تنفيذ التعليمة الثانية (٢) . وفي كلتا الحالتين يتم تنفيذ تعليمة السطر الذي يليه تعليمة إذا .

ونشير هنا إلى إمكانية وجود أكثر من تعليمة بعد كلمة THEN وكلمة ELSE بشرط أن يفصل بينهما «:» وذلك كما هو موضح أدناه :

10 IF A > B THEN التعليمة الثانية : التعليمة الأولى ELSE

لاحظ أن عدد الحروف التى يمكن أن تظهر فى رقم سطر واحد هو ٢٥٦ حرفاً ، وهذا يحد من عدد التعليمات التى يمكن أن تظهر فى رقم سطر واحد . لذلك فى حالة مثل هذه يفضل تقليل الفراغ بين التعليمات قدر الإمكان .

ويمكن تعديل البرنامج السابق وفقاً لهذا الأسلوب كالتالى :

(أ) إذا كان عدد ساعات العمل أقل أو يساوى ٤٠ ساعة فذلك يعنى أن الموظف لم يعمل خارج دوام ، وعليه يكون :

خارج الدوام = صفراً
LET O = 0

والراتب الإجمالى = عدد ساعات العمل × أجره الساعة

LET N = H * P

(ب) أما إذا كان عكس ذلك (ELSE) أى لم يتحقق الشرط ، فيعنى أن الموظف قد عمل خارج دوام ، وعليه يكون :

خارج الدوام = (عدد ساعات العمل — ٤٠) × (أجره الساعة × ١,٥)

LET O = (H - 40) * (P * 1.5)

والراتب الإجمالى = خارج الدوام + (٤٠ × أجره الساعة)

LET N = O + (40 * P)

يمكننا تنفيذ الخطوتين أ ، ب باستخدام العبارة التالية :

(أ)
IF H <= 40 THEN O = 0 : N = H * P
ELSE

(ب)
O = (H - 40) * (P * 1.5) : N = O + (40 * P)

وبإجراء هذه التعديلات ، تصبح صيغة البرنامج كالتالى :

شكل (٨-١ ب)

برنامج لحساب الرواتب الإجمالية لموظفي مؤسسة معينة باستخدام تعليمة IF / THEN / ELSE

```

10 REM استدعاء البرنامج الفرعي
20 GOSUB 40
30 END
40 REM لطباعة المناوين
50 PRINT "      NAME      HOURS WORKED      RATE      BASE SALARY      OVERTIME      NET
PAY"
60 PRINT "=====
=====
=====
=====
=====
=====
70 REM اسناد قيم اوليه للمتغيرات
80 LET TY = 0: LET T1 = 0
90 REM لقراءة اسم الموظف، عدد ساعات العمل، اجرة الساعه
100 READ N$,H,P
110 REM للاستفسار عن نهاية البيانات
120 IF N$ = "dummy" THEN 300
130 REM للاستفسار من وجود خارج دوام
140 IF H <= 40 THEN N = H * P : O = 0
ELSE O = (H - 40) * (P * 1.5) : N = 0 + (P * 40)
150 REM لطباعة اسم الموظف، عدد ساعات العمل، اجرة الساعه، خارج دوام، راتب اجمالي
160 REM للايجاد الراتب الاساسي
170 LET B = N - O
180 LET T = T + N
190 LET T1 = T1 + O
200 PRINT TAB(2);N$; TAB(20);H; TAB(37);P; TAB(48);B; TAB(58);O;TAB(67);N
210 PRINT "-----
-----
220 REM لقراءة اسم الموظف، عدد ساعات العمل، اجرة الساعه
230 READ N$,H,P
240 GOTO 120
250 DATA "ALI AHMED",60,23
260 DATA "SALIM ALI",42,25
270 DATA "MOHAMED OMER",36,19
280 DATA "AHMED HASAN",52,18
290 DATA "dummy",0,0
300 PRINT TAB(2);N$;
=====
=====
310 REM لطباعة اجمالي خارج الدوام و اجمالي الراتب للموظفين
320 PRINT TAB(40);"t o t a l s";TAB(57);T1;TAB(66);T
330 PRINT TAB(2);N$;
=====
=====
340 RETURN

```

وفي حالة تنفيذ البرنامج في الشكل السابق ستكون المخرجات على الشكل التالي :

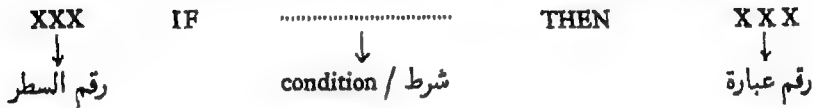
NAME	HOURS WORKED	RATE	BASE SALARY	OVERTIME	NET PAY
=====	=====	=====	=====	=====	=====
ALI AHMED	60	23	920	890	1610
SALIM ALI	42	25	1000	75	1075
MOHAMED OMER	36	19	684	0	684
AHMED HASAN	52	18	720	324	1044
-----	-----	-----	-----	-----	-----
t o t a l s				1089	4413
=====					

نلاحظ في هذا البرنامج كيفية استخدام أكثر من عبارة في نفس السطر باستخدام الرمز «:» ، كما هو مبين في العبارة 110 بالإضافة إلى استخدام تعليمة (عكس ذلك ELSE) للتخلص من الاستخدام المتعدد لتعليمة إذا/إذن IF/THEN .

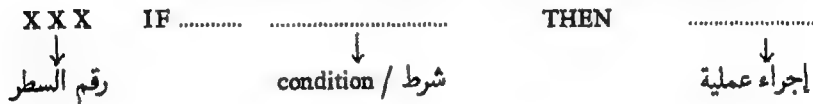
ملخص ما تقدم عن استخدامات تعليمة IF/THEN :

لقد أمكن التحكم في سير تنفيذ البرنامج باستخدام تعليمة إذا/إذن IF/THEN في عدة طرق وهي :

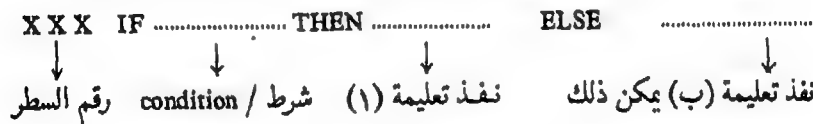
١ — تحويل سير التنفيذ في حالة تحقيق الشرط إلى رقم عبارة في البرنامج



٢ — تنفيذ تعليمة (إدخال / إخراج / معالجة) في حالة تحقيق الشرط .

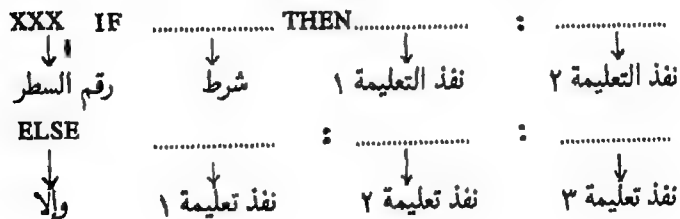


٣ — تنفيذ عملية (أ) في حالة تحقيق الشرط وإلا (عكس ذلك) تنفيذ عملية (ب) .



٤ — تنفيذ أكثر من عملية في عبارة واحدة في حالة تحقيق الشرط وإلا (عكس ذلك)

نفذ العمليات التالية في العبارة التالية لتعليمة ELSE



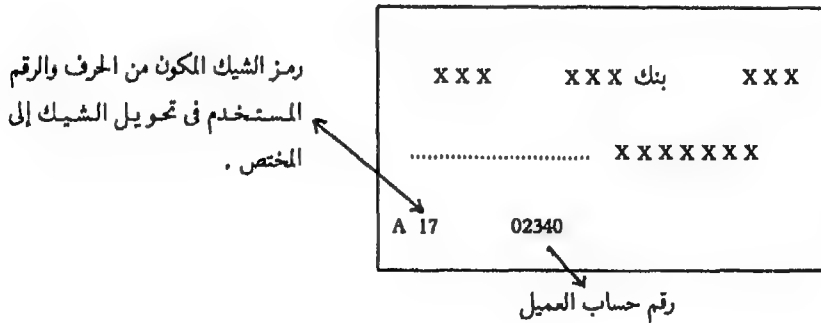
تطوير برنامج باستخدام تعليمة IF/THEN والمقارنات OR , AND :

تم استخدام تعليمة IF/THEN فيما سبق في وضع يتم فيه التحكم بمسار تنفيذ خطوات البرنامج ، بناء على نتيجة مقارنة أحادية بين قيمتي متغيرين ، ولكن في بعض الحالات قد تعتمد نتيجة التشعب على تحقيق أكثر من شرط (مقارنة) بين أكثر من زوج من المتغيرات .

ففي هذه الحالة تظهر هذه الشروط (المقارنات) بين كلمتي IF و THEN وسنعطى هنا مثالا على كيفية استخدام عبارة IF/THEN مع المقارنات المتعددة لتحقيق التشعب إلى التعليمة الظاهرة بعد THEN

مثال (٨ - ٢) :

يعمل بنك معين على تسلم الشيكات ، ومن ثم توزيعها على ثلاثة موظفين بناء على رمز مكتوب على هذه الشيكات ، ويظهر هذا الرمز قبل رقم حساب الزبون ، ويتكون هذا الرمز من جزأين ، الأول : حرفي يتكون من أحد الأحرف التالية : B , C , D , A ، والثاني : رقمي مكون من خانتين وذلك حسب الشكل التالي :



- (أ) فإذا كان الحرف الأول يساوى أيّاً من الرموز التالية A , B أو C يتم تحويله للموظف x وذلك بغض النظر عن الجزء الرقعى .
- (ب) إذا كان الحرف الأول يساوى D والرقم الذى يليه يساوى 14 يتم تحويله للموظف y .
- (ج) إذا كان الحرف الأول يساوى D والرقم الذى يليه لا يساوى 14 يحول للموظف x .
- (د) إذا كان الحرف الأول لا يساوى A , B , C أو D يعتبر هذا الشيك غير سليم ، وتتم إعادته إلى المصدر .

الهدف : الحصول على تقرير فى نهاية الدوام يحتوى على التالى :

- ١ — عدد الشيكات المحولة للموظف X
- ٢ — عدد الشيكات المحولة للموظف y
- ٣ — عدد الشيكات غير السليمة (المعادة) .

أولاً — خطوات الحل :

- ١ — الحصول على رمز الشيك والرمز الذى يليه .
- ٢ — الاستفسار عن رمز الشيك إذا كان يساوى A أو B أو C أضف واحداً للعدد الخاص بالموظف X .
- ٣ — الاستفسار عن رمز الشيك إذا كان يساوى D والرمز الذى يليه يساوى 14 أضف واحداً للعدد الخاص بالموظف y .
- ٤ — الاستفسار عن رمز الشيك إذا كان يساوى D والرمز الذى يليه لا يساوى 14 أضف واحداً للعدد الخاص بالموظف X .
- ٥ — الاستفسار عن رمز الشيك إذا كان لا يساوى A أو B أو C أو D أضف واحداً للعدد الخاص بالشيكات غير السليمة (المعادة) .

نلاحظ أن كلاً من الخطوات ٢ ، ٣ ، ٤ ، ٥ ، يتطلب منا القيام بأكثر من عملية مقارنة . وسنتعرض الآن للتعليمات الممكن استخدامها في لغة بيسك لمثل هذا النوع من عمليات المقارنة المستخدمة في تعليمة IF ... THEN

١ - تعليمة «أو» (OR) وتستخدم للاستفسار عن تحقيق أى من المقارنات ، كما هو مطلوب في الخطوة ٢ ، حتى تتم زيادة عداد الموظف x بواحد ، وتظهر هذه التعليمة كالتالى :

الشكل العام لتعليمة «أو» (OR)										
X X X	IF	X	=	X X	OR	X	=	X X	THEN	X X X X
↓		↓	↓	↓	↓	↓	↓	↓	↓	↓
رقم السطر		اسم	رمز	قيمة	تعليمة	اسم	رمز	قيمة	إذن	تعليمة اقرأ أو اطبع أو أسند أو الذهاب إلى رقم عبارة أو استدعاء برنامج فرعى .
			المقارنة	المتغير	أو		المقارنة	المتغير		

وعليه فيمكننا تمثيل الخطوة (٢) بالعبارة التالية :

X X X IF A\$ = «A» OR A\$ = «B» OR A\$ = «C» THEN N1 = N1 + 1
حيث نضيف واحداً للعداد N1 في حالة أن يكون المتغير A\$ يساوى أياً من القيم «A» أو «B» أو «C»
إذن في حالة استخدام تعليمة OR يجب تحقيق واحد فقط من الشروط (عمليات المقارنة) حتى يتم تنفيذ تعليمة ما بعد THEN .

٢ - تعليمة «و» (AND) وتستخدم للتأكد من تحقيق جميع عمليات المقارنة ، كما هو مطلوب في الخطوة ٣ حتى تتم زيادة عداد الموظف Y بواحد ، وشكل هذه التعليمة كالتالى :

الشكل العام لتعليمة «و» AND										
XXX	IF	X	=	XX	AND	X	=	XX	THEN	XXXXX
↓		↓	↓	↓	↓	↓	↓	↓		↓
رقم السطر		اسم	رمز	قيمة	تعليمة	اسم	رمز	قيمة		عملية أو
		المتغير	المقارنة	المتغير	و	المتغير	المقارنة	المتغير		أمر اذهب إلى رقم
										عبارة أو أمر استدعاء
										برنامج فرعى

• يمكن أن يكون رمز المقارنة أيًا من الرموز التي تم شرحها

وعليه يمكننا تمثيل الخطوة (٣) بالعبارة التالية :

IF A\$ = «D» AND = 14 THEN N2 = N2 + 1

أى نضيف واحدًا للعدد N2 في حالة أن يكون المتغير A\$ مساوياً N للقيمة D وقيمة المتغير B تساوى 14

إذن في حالة استخدام تعليمة «و» (AND) يجب تحقيق جميع الشروط ، (عمليات المقارنة) حتى يتم تنفيذ تعليمة ما بعد كلمة THEN

٣- ويمكننا تمثيل كل من :

أ) الخطوة (٤) بالعبارة التالية :

IF A\$ = «D» AND B < > 14 THEN N1 = N1 + 1

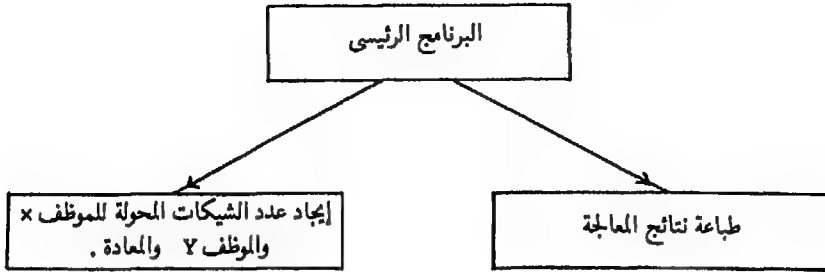
أى إضافة واحد للعدد N1 في حالة أن يكون المتغير A\$ يساوى D والمتغير B لا يساوى 14

ب) الخطوة (٥) بالعبارة التالية :

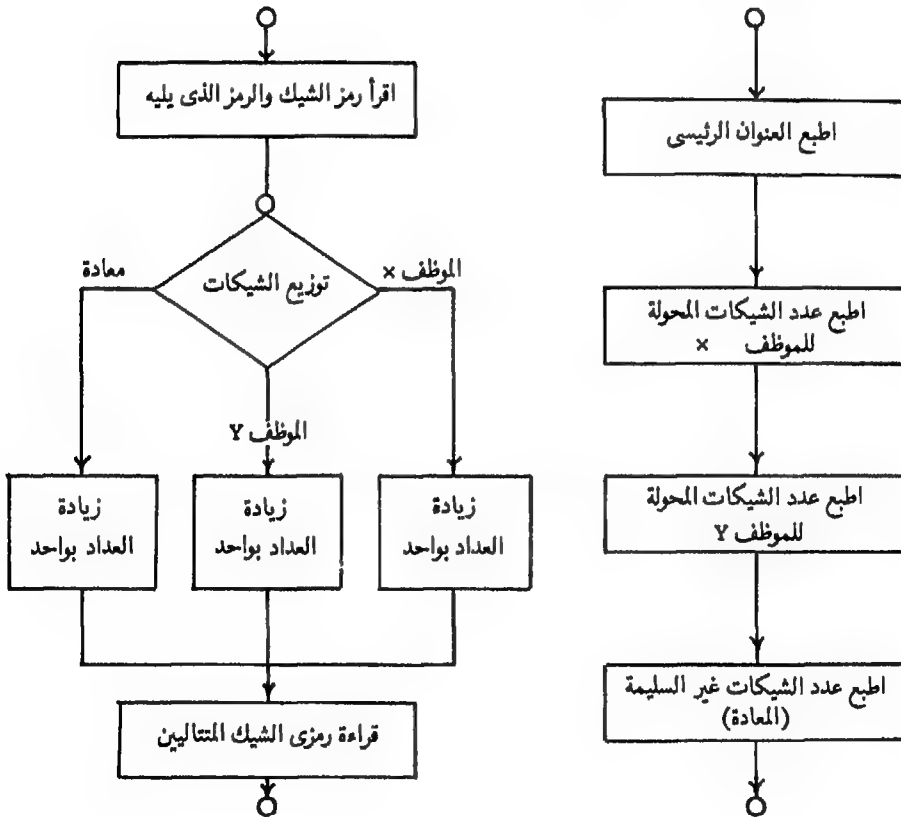
IF A\$ < > «A» A A\$ < > «B» A\$ < > «C» A\$ < > «D»
THEN N3 = N3 + 1

أى إضافة واحد للعدد N3 في حالة أن يكون المتغير A\$ لا يساوى أيًا من A أو B أو C أو D

ثانياً - الهيكل الهرمي :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (٨ - ٢)

برنامج لإيجاد عدد الشيكات المحولة للموظف x والموظف y والمعاد

```

10 REM البرنامج الرئيسي
20 REM استدعاء برنامج فرعي
30 GOSUB 100
40 REM استدعاء برنامج فرعي لطباعة النتائج
50 GOSUB 290
60 END
100 REM برنامج فرعي لحساب عدد الشيكات
110 REM اسناد القيمة صفر لعدادات الحالات الثلاثة
120 LET N1=0: LET N2 = 0: LET N3 = 0
130 REM قراءة رمز الشيك و الرمز الذي يليه
135 READ A$,B
140 REM هل نهاية البيانات
150 IF A$ = "0" THEN ?70
155 REM x هل الشيك ، من الموظف
160 IF A$ = "a" OR A$ = "b" OR A$ = "c" THEN N1 = N1 + 1
170 IF A$ = "d" AND B = 14 THEN N2 = N2 + 1
180 REM y هل الشيك يخص الموظف
190 IF A$ = "d" AND B <> 14 THEN N1 = N1 + 1
200 REM هل الشيك غير سليم
210 IF A$ <> "a" AND A$ <> "b" AND A$ <> "c" AND A$ <> "d" THEN N3 = N3 + 1
220 REM لقراءة رمز الشيك و الرمز الذي يليه
230 READ A$,B
240 GOTO 140
250 DATA a,12,b,20,c,18,d,5,d,14,a,14,d,14,d,14,a,13
260 DATA b,15,d,14,n,14,r,14,d,14,c,15,d,16,0,0
270 RETURN
290 REM برنامج فرعي لطباعة النتائج
300 PRINT "=====
310 PRINT "          "N1," x عدد الشيكات المحولة للموظف
320 PRINT "          "N2," y عدد الشيكات المحولة للموظف
330 PRINT "          "N3," عدد الشيكات المعادة
340 PRINT "=====
350 RETURN

```

وعند تنفيذ البرنامج ، ستظهر النتائج كالتالي :

```

=====
9      x عدد الشيكات المحولة للموظف
5      y عدد الشيكات المحولة للموظف
2      عدد الشيكات المعادة
=====

```

أساليب تمثيل المقارنات المركبة :

هناك أساليب لتمثيل المقارنات المتعددة بالرسم حتى يسهل فهمها وبرمجتها ، مثل جدول القرارات وشجرة القرارات ، كما هو موضح بالرسمين التاليين :

(أ) جدول القرارات : وينقسم إلى قسمين ، الأول : تسرد به الاستفسارات وحالاتها (نعم أولا) ، والثاني : القرارات التي ستؤخذ على ضوء حالات هذه الاستفسارات ، ففي مثالنا يكون الجدول كالتالى :

حالاتها				الاستفسارات	IF
لا	لا	لا	نعم	رمز الشيك يساوى A أو B أو C	
لا	نعم	نعم	لا	رمز الشيك يساوى D	
	لا	نعم		الرقم الذى يلى الرمز يساوى 14	
حتى يتم تنفيذها				القرارات	THEN
	✓		✓	يحول للموظف x	
		✓		يحول للموظف Y	
✓				غير سليم و يعاد	

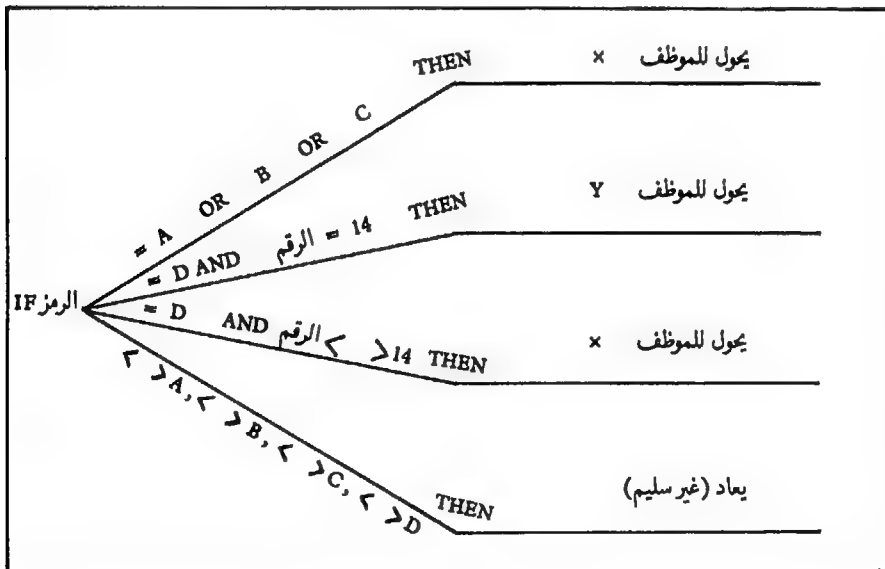
فكما نرى من الجدول :

١ - يتم تحويل الشيكات للموظف x في حالتين : أ) إذا كان الرمز يساوى A أو B أو C (نعم) . ب) إذا كان رمز الشيك يساوى D (نعم) ، والرقم الذى يلي الرمز لا يساوى 14 .

٢ - يتم تحويل الشيكات للموظف y عندما يبين الاستفسار عن رمز الشيك أنه يساوى D (نعم) ، والرقم الذى يلي الرمز يساوى 14 (نعم) .

٣ - تتم إعادة الشيك (غير سليم) عندما يبين الاستفسار عن رمز الشيك أنه يساوى A أو B أو C (لا) وعندما يبين الاستفسار عن رمز الشيك أنه يساوى D (لا) .

ب) شجرة القرار : وتنقسم فروعها إلى عدد الحالات التى يراد الاستفسار عنها كما هو مبين فى الشكل التالى :

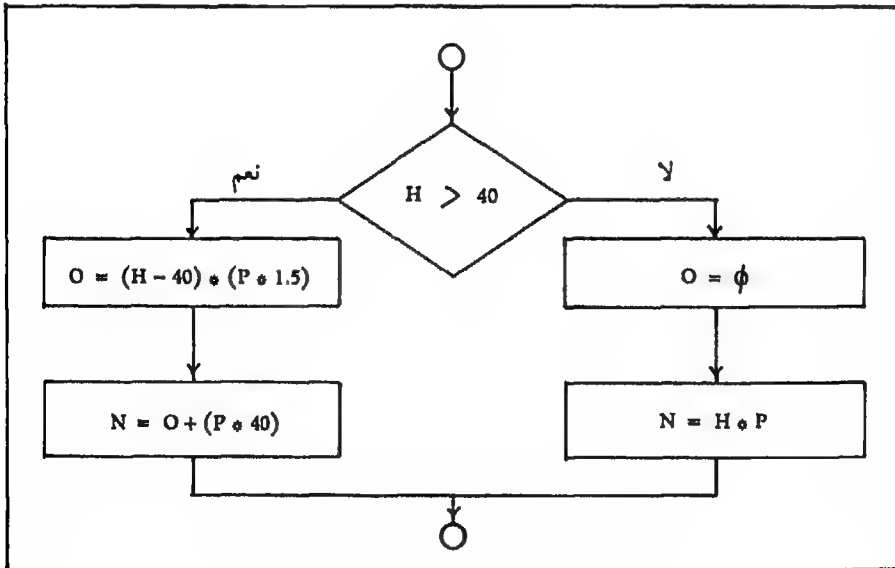


التشعب المتعدد

تطوير برنامج بواسطة تعليمة التشعب المتعدد ON / GOTO

بناء على قيمة معطاة :

نحدثنا في الجزء السابق عن كيفية التحكم في سير تنفيذ البرنامج باستخدام تعليمة إذا/إذن IF/THEN ، حيث إن تسلسل تنفيذ البرنامج قد تفرع حسب نتيجة المقارنة بين القيمة التي يحويها المتغير وقيمة ثابتة ، كما هو الحال في شكل (٨ - ١) عبارة (180) ، ويعتبر هذا النوع من التفرع المشروط بنى حدين (مسارين) ، كما هو موضح في الشكل التالي :



حيث إنه طبقاً للقيمة التى يحتوئها المتغير H ، فإن سير تنفيذ البرنامج سيتفرع إما لإيجاد المبلغ المستحق من خارج الدوام ، ومن ثم الراتب الإجمالى فى حالة كون قيمة H أكبر من ٤٠ ساعة أسبوعية ، أو لإيجاد المبلغ الإجمالى دون خارج الدوام فى حالة كون قيمة H تساوى أو أقل من ٤٠ ساعة أسبوعية ، وفى هذه الحالة الأخيرة يكون مبلغ خارج الدوام = صفراً . وتضاف العبارة الأخيرة لجعل المقارنة بين المسارين أكثر وضوحاً وتمائلاً .

ولكن يلاحظ أنه فى بعض الحالات يتطلب تحقيق هدف معين التشعب المتعدد ، بناء على قيمة معطاة لمتغير معين . ومن الأمثلة على هذه الحالة معالجة بيانات تابعة لقطع مختلفة ، أو لمناطق جغرافية متعددة ، أو لآلات إنتاجية ذات وظائف مختلفة ، أو لتبويب البيانات فى أصناف محددة ... وهكذا .

ولتسهيل عملية إجراء مثل هذه المقارنات التى تؤدى إلى التشعب المتعدد ، توجد تعليمة ON/GOTO ونشرح كيفية استخدام هذه التعليمة فى المثال التالى :

مثال (٨ - ٣) :

الهدف : رجل مبيعات يوجد لديه صنفان من البضاعة ، ويريد معرفة المبلغ الإجمالى لمبيعاته فى نهاية الأسبوع لكلا الصنفين اللذين يبيعهما ، مع العلم بأن كل قطعة من الصنف الأول تباع بمبلغ (١٥٠) ريالاً ، ومن الصنف الثانى تباع بمبلغ (١٧٥) ريالاً .

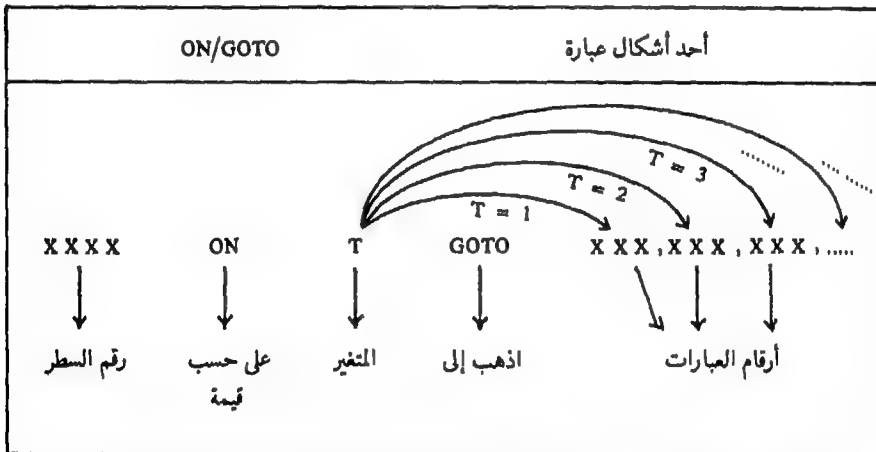
أولاً - خطوات الحل :

- ١ - الحصول على عدد القطع المباعة ورمز كل صنف .
- ٢ - إذا كان الرمز يساوى ١ نوجد الإجمالى لمبيعات الصنف الأول لذلك اليوم ، ومن ثم نضيفه إلى إجمالى مبيعاته .

٣- إذا كان الرمز يساوى ٢ نوجد الإجمالى لمبيعات الصنف الثانى لذلك اليوم ، ومن ثم نضيفه إلى إجمالى مبيعاته .

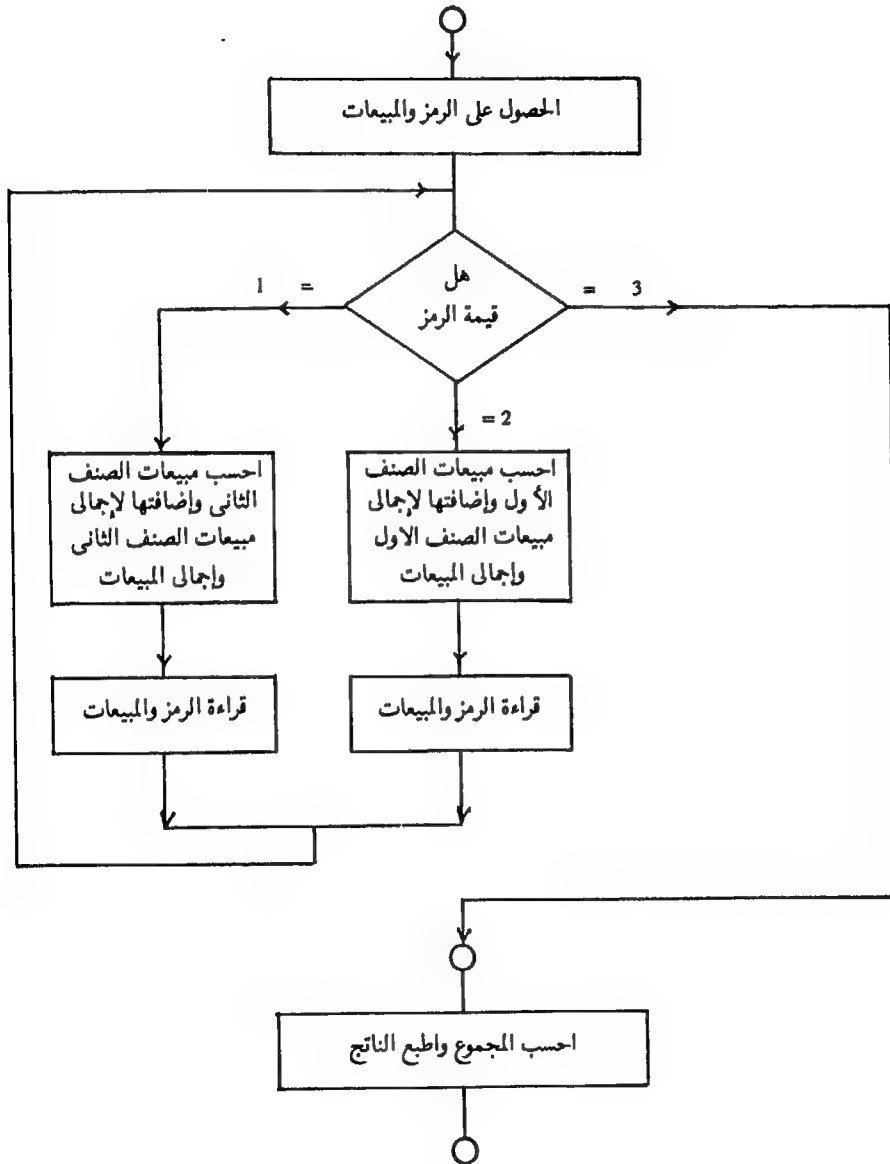
٤- إذا كان الرمز يساوى ٣ نوجد المبلغ الكلى لمبيعات الصنفين ومن ثم طباعة النتائج والتوقف .

نلاحظ من هذا المثال أن القيمة المراد الاستفسار عنها تتراوح ما بين ١ - ٣ ، وهذا يتطلب منا القيام بأكثر من عملية مقارنة أى تكرار استخدام IF/THEN ، ولكن استخدام تعليمة ON/GOTO سيمكننا من الاستغناء عن ضرورة تكرار عبارة IF/THEN ، وذلك بالتشعب إلى ثلاثة أرقام عبارات مختلفة بناء على صنف المبيعات ، ويكون أحد أشكال هذه التعليمة كالتالى :



و يتم الانتقال إلى أرقام هذه العبارات حسب قيمة المتغير T . كما هو واضح من الأسهم الموصلة بين المتغير T وأرقام العبارات الظاهرة بعد كلمة GOTO ، فإذا كانت القيمة تساوى ١ يتم الذهاب إلى رقم العبارة الأولى الذى يلي GOTO ، وإذا كانت القيمة تساوى ٢ يتم الانتقال إلى رقم العبارة الثانية الذى يلي GOTO ، وهكذا .

ثانياً وثالثاً – سنستخدم برنامجاً فرعياً واحداً لهذا المثال . ويمثل الشكل التالي رسم البرمجة التركيبية لهذا الجزء :



رابعاً - يحتوى الشكل التالى على البرنامج الكامل لتحقيق الهدف من هذا المثال :

شكل (٨ - ٣)

برنامج لإيجاد المبلغ الإجمالى فى نهاية الأسبوع لمبيعات صنفين من البضاعة

```

10 REM      استخدماء برنامج فرعى لحساب المبلغ الا جمالى
20 GOSUB 40
30 END
40 REM      اسناد القيمة صفر كقيمة اوليه للمجموع
50 LET T1= 0
60 LET T2 = 0
70 REM      قراءة الصنف و عدد القطع المباعه
80 READ T,Q
90 REM      الاستفسار عن نوع الصنف للتخمين فى سير تنفيذ العمليات
100 ON T GOTO 110,160,210
110 REM      للايجاد اجمالى مبيعات اليوم من الصنف الا ول و اضافته للمجموع الكلى
120 LET A = Q * 150
130 LET T1 = T1 + A
140 READ T,Q
150 GOTO 100
160 REM      للايجاد اجمالى مبيعات اليوم من الصنف الثانى و اضافته للمجموع الكلى
170 LET A = Q * 175
180 LET T2 = T2 + A
190 READ T,Q
200 GOTO 100
210 REM      للايجاد المبلغ الا جمالى للمبيعات من الصنفين وطباعة النتائج
220 LET S = T1 + T2
230 PRINT "=====
240 PRINT "المبلغ الا جمالى لمبيعات الصنف الا ول = "T1;"ريالا سعوديا "
250 PRINT "المبلغ الا جمالى لمبيعات الصنف الثانى = "T2;"ريالا سعوديا "
260 PRINT "=====
270 PRINT "المبلغ الا جمالى لمبيعات الا اسبوع من الصنفين = "S;"ريالا سعوديا "
280 PRINT "=====
290 DATA 1,30,2,10,1,0,2,5,1,15,2,9,1,8,2,6,1,40,2,21
300 DATA 1,18,2,13,1,7,2,3,3,0
310 RETURN

```

وعند تنفيذ البرنامج ستظهر النتائج كالتالى :

```

=====
المبلغ الا جمالى لمبيعات الصنف الا ول = 17700 ريال سعوديا
المبلغ الا جمالى لمبيعات الصنف الثانى = 11725 ريال سعوديا
=====
المبلغ الا جمالى لمبيعات الا اسبوع من الصنفين = 29425 ريال سعوديا
=====

```

تطوير برنامج باستخدام تعليمة ON/GOTO
مع إجراء عمليات حسابية للتوصل إلى قيم التفرع :

في بعض الحالات ستكون القيم التي سيحويها المتغير أكبر بكثير من القيم التي مثلها كل من المتغير T في شكل (٨ - ٣) أو المتغير N في شكل (٨ - ٤) وهذا يوجب علينا وضع أرقام عبارات مساوية للقيمة العليا التي سيحتويها هذا المتغير .
وللتحليل على مثل هذا النوع من القيم يمكننا القيام بعمليات حسابية على هذا المتغير، لجعل هذه القيمة تتناسب مع نوع العمليات المراد إجراؤها، وذلك كما في المثال التالي :

مثال (٨ - ٤) :

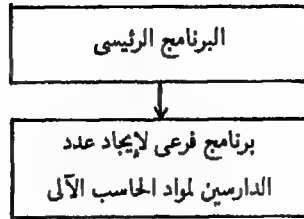
الهدف : قسم الحاسب الآلى في معهد الإدارة العامة يرغب في إيجاد عدد الدارسين الذين انضموا لكل من المواد الأربع التالية :

رمزها	اسم المادة
١٠١	مقدمة في الحاسب الآلى
١٠٢	البرمجة بلغة البيسك
١٠٥	التحليل والتصميم
١٠٦	إدارة المشاريع

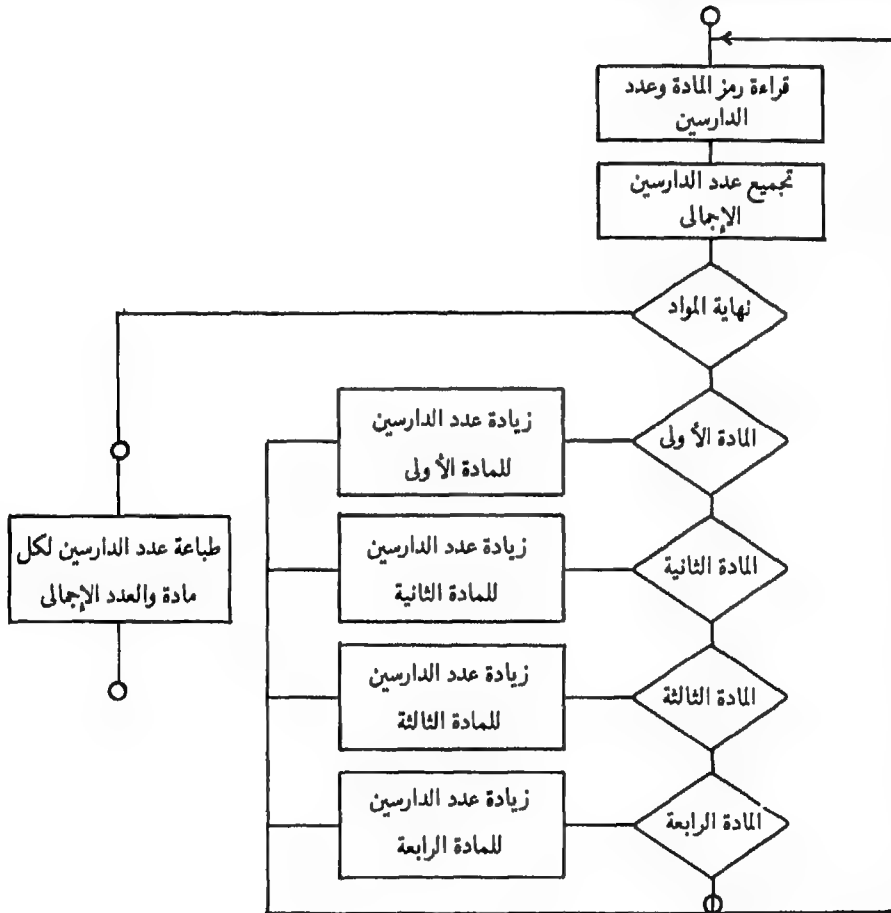
أولاً - خطوات الحل :

- ١ - الحصول على رمز المادة وعدد الدارسين .
- ٢ - إضافة عدد الدارسين لمجموع المادة .
- ٣ - إضافة عدد الدارسين للمجموع الكلى للدارسين .

ثانياً : الهيكل الهرمي للبرنامج :



ثالثاً : مخطط البرمجة التركيبية :



رابعاً - البرنامج كاملاً في الشكل التالي :

شكل (٨ - ٤)

برنامج لإيجاد عدد الدارسين لمواد الحاسب الآلي	
10 REM	استدعاء برنامج فرعي
20 GOSUB 40	
30 END	
40 REM	برنامج فرعي لايجاد عدد الدارسين المنضمين لمواد الحاسب الآلي
50 REM	قراءة رمز المادة وعدد الدارسين
60 READ N,S	
70 REM	للاستفسار عن نهاية البيانات
80 IF N=0 THEN 240	
90 T=T+S	
100 ON N-100 GOTO 110,140,60,60,170,200	
110 REM	إضافة عدد الدارسين للعدد الإجمالي للمادة الإولي
120 N1=N1+S	
130 GOTO 80	
140 REM	إضافة عدد الدارسين للعدد الإجمالي للمادة الثانية
150 N2=N2+S	
160 GOTO 80	
170 REM	إضافة عدد الدارسين للعدد الإجمالي للمادة الثالثة
180 N3=N3+S	
190 GOTO 80	
200 REM	إضافة عدد الدارسين للعدد الإجمالي للمادة الرابعة
210 N4=N4+S	
220 GOTO 80	
230 DATA 101,5,102,7,103,6,104,6,101,4,102,12,105,12,106,14,0,0	
240 REM	طباعة المخرجات
250 PRINT "	عدد الدارسين الذين انضموا الى المواد الاربع التالية
260 PRINT "	=====
270 PRINT	المادة " TAB(49); "عدد الدارسين" TAB(5); "
280 PRINT	TAB(5); " TAB(49); "=====
290 PRINT	TAB(9); N1; TAB(30); "مقدمه في معالجة البيانات
300 PRINT	TAB(9); N2; TAB(37); "البرمجه بلغة بيك
310 PRINT	TAB(9); N3; TAB(38); "التحليل والتصميم
320 PRINT	TAB(9); N4; TAB(40); "ادارة المشاريع
330 PRINT	=====
340 PRINT	TAB(9); T; TAB(17); "العدد الإجمالي للدارسين =
350 PRINT	=====
360 RETURN	

وفي حالة تنفيذ البرنامج ، تظهر النتائج كالتالي :

عدد الدارسين الذين انضموا الى المواد الاربع التاليه	
=====	
الماده	عدد الدارسين
-----	-----
مقدمه في معالجة البيانات	9
البرمجه بلغة بيك	19
التحليل والتصميم	12
ادارة المشاريع	14
=====	=====
العدد الإجمالي للدارسين =	68
=====	=====

لو تتبعنا البرنامج السابق نجد أن القيم التي سيحتويها المتغير تتراوح ما بين ١٠١ إلى ١٠٧ ، وهذا يعنى أننا بحاجة إلى وضع مائة وسبعة أرقام للعبارات تلى GOTO ، لكن يمكننا تقليل ذلك الرقم بإجراء عملية الطرح على هذه القيمة دون أن تؤثر على منطق عمل البرنامج ، وذلك بطرح (١٠٠) من كل قيمة تسند للمتغير ، وهذا ماتم في العبارة (١٥٠) . فالقيمة المستخدمة في المقارنة للتعبير الجبرى هى ١ ، ٢ ، ٣ ، ٤ ، ٥ ، ٦ ، ٧ وهى مقابلة لرموز المواد ١٠١ ، ١٠٢ ، ١٠٣ ، ١٠٤ ، ١٠٥ ، ١٠٦ ، ١٠٧ (والأخير يعتبر مؤشراً لنهاية البيانات) . كما أن البيانات في العبارة (٢٨٠) احتوت على رمزى مادتى ١٠٣ ، ١٠٤ وهذان الرمزان خاصان بمادتين أخريين لعللاقة لهما بما هو مطلوب منا القيام به ، فيتم إهماله لتوجيه الحاسب للقراءة في كل مرة وجدت بها قيمة التغير تساوى ٣ أو ٤ دون أى تعداد .

وكما أمكننا طرح (100) من قيمة المتغير N يمكننا إضافة قيمة ثابتة للمتغير ، في حالة حصوله على قيم سالبة ، أو ضربه في قيمة ثابتة ، أو رفعه إلى قوة ثابتة . أما في حالة القسمة إذا حصلت كسور عشرية في القيمة التى ستسند للمتغير فيتم إهمالها ، وسيسند الرقم الصحيح فقط . فمثلا إذا كانت قيمة N (٢٩) وتمت قسمتها على القيمة (١٠) في المثال التالى :

ON N / 10 GOTO 200 , 250 , 370

يتم الذهاب إلى العبارة رقم ٢٥٠ أى تم إهمال (٩) الكسر العشرى وإسناد القيمة (٢) للمتغير N .

وهذه بعض الأمثلة من عبارة ON/GOTO توجد بها تعبيرات جبرية مختلفة :

ON X - W 10 GOTO

ON 11 * 12 - 100 GOTO

ON (1 ↑ 2 - J)/20 GOTO

كيفية استخدام تعليمة اذهب إلى

برنامج فرعى والعودة منه ON/GOSUB , RETURN : ٢

إذا تتبعنا خطوات تنفيذ البرنامج شكل (٨ - ٣) نجد أن العبارة (٨٠) تعمل على قراءة عدد القطع المباعة ومن ثم صنف القطع ، وعبارة (100) تستفسر عن نوع الصنف (المتغير T) فإذا كانت قيمة المتغير T تساوى ١ يتم الذهاب إلى عبارة (110) لإيجاد المبلغ الإجمالى لمبيعات اليوم من الصنف الأول ، وإذا كانت قيمة المتغير T تساوى ٢ يتم الذهاب إلى عبارة (160) لإيجاد المبلغ الإجمالى لمبيعات اليوم من الصنف الثانى ، أما إذا كانت قيمة المتغير T تساوى ٣ فيتم الذهاب إلى عبارة (210) لإيجاد المبلغ الإجمالى للمبيعات من الصنفين ومن ثم تتم طباعة النتائج .

وهناك تعليمة أخرى مشابهة لتعليمة ON/GOTO وهى تعليمة ON/GOSUB ، حيث يتم التفرع إلى أرقام العبارات التى تحدد بدايات برامج فرعية بدلاً من أرقام عبارات فى البرنامج الرئيسى ، وذلك بناء على قيمة متغير معين كما فى التعليمة السابقة .

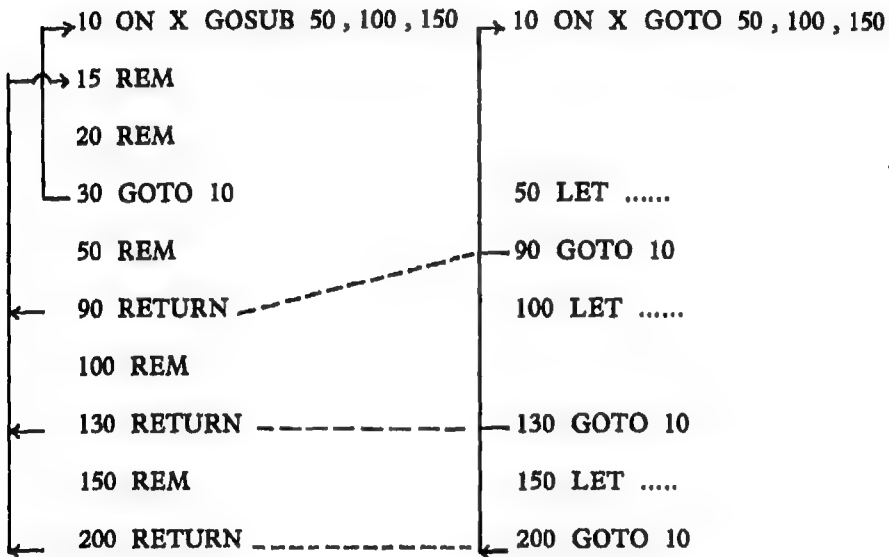
لقد أمكننا التحكم فى تسلسل تنفيذ البرنامج فى شكل (٨ - ٣) باستخدام تعليمة ON/GOTO فى العبارة رقم (100) وذلك لتنفيذ أى من العبارات 110 , 160 , 210 اعتماداً على القيمة المخزنة فى المتغير T (١ ، ٢ ، ٣) .

أما إذا استخدمت عبارة ON/GOSUB فإن أرقام العبارات 110 , 160 , 210 ستكون بدايات لبرامج فرعية . وحال الانتهاء من تنفيذ البرنامج الفرعى يتم العودة إلى السطر الذى يلي عبارة ON/GOSUB باستخدام تعليمة RETURN فى كل من البرامج الفرعية . لذلك تستخدم تعليمتا ON/GOSUB و RETURN للتفرع من الأجزاء المتحكممة (البرامج الرئيسية) إلى الأجزاء المنفذة (البرامج الفرعية) والعودة إليها . والشكل العام لهذه العبارة هو كالتالى :

ON/GOSUB		الشكل العام لعبارة
<pre> X X X ↑ رقم السطر </pre>	<pre> X ↑ متغير </pre>	<pre> GOSUB X X X , X X X , X X X , ... ↙ ↗ ↘ ↘ أرقام أسطر العبارات الأولى من البرامج الفرعية </pre>

مقارنة تعليمتى ON/GOSUB و ON/GOTO :

و يوضح المثال التالى الفرق بين تعليمتى ON/GOSUB و ON/GOTO



لاحظ الفرق التالى بين التعليمتين : فى حالة ON/GOTO يتم العودة إلى تنفيذ العبارة مرة أخرى بعد التشعب منها عن طريق تعليمة GOTO . أما فى التعليمة الأخرى فيتم العودة إلى السطر الذى يلي بواسطة تعليمة RETURN ، ومن ثم تستخدم تعليمة GOTO لتنفيذ عبارة ON/GOSUB مرات أخرى .

ملاحظات على استخدامات تعليمتى ON / GOTO و ON / GOSUB :

١ — يجب أن يكون عدد أرقام أسطر العبارات التى تلى GOTO ، مساوياً للقيمة العظمى التى سيحويها المتغير X .

٢ — فى حالة استخدام ON X GOSUB يجب أن يكون عدد أرقام بداية البرامج الفرعية ، التى تلى GOSUB مساوياً للقيمة العظمى ، التى سيحويها المتغير X .

٣ — يفضل استخدام التعليمات ON X GOTO أو ON X GOSUB عندما تكون القيمة العظمى للمتغير X معروفة مسبقاً ، وأن تكون القيمة العظمى التى سيحويها المتغير X لا تزيد على عشرة (١٠) .

٤ — يمكننا استخدام رقم عبارة أو برنامج فرعى لأكثر من قيمة للمتغير X .

مثال أ — `ON X GOTO 100 , 150 , 170 , 100 , 200`

هنا يتم تنفيذ العبارة (١٠٠) عندما تكون قيمة X تساوى (١) أو (٤) .

مثال ب — `ON N GOSUB 100 , 300 , 400 , 100 , 700`

هنا يتم استدعاء البرنامج الفرعى الذى يبدأ من العبارة (١٠٠) ، عندما تكون قيم المتغير X تساوى (١ أو ٤) .

٥ — فى حالة حصول المتغير X على قيمة أعلى من عدد أرقام العبارات سواء كانت تابعة

لـ GOTO أو GOSUB ، فإن الحاسب إما يعمل على تنفيذ العبارة التالية للعبارة

..... ON X ، أو ينهى تنفيذ البرنامج حسب طبيعة الحاسب المستخدم (ارجع

للفصل السابع عشر) .

٦ — يمكن استخدام ON/GOTO فى حالات التشعب داخل الجزء الواحد فى الهيكل

المهرمى ، وخاصة إذا كانت المعالجات التابعة لكل منها بسيطة . أما فى حالة

التشعب لتنفيذ أجزاء متكاملة فى الهيكل المهرمى للبرنامج فيمكن استخدام

ON/GOSUB .

تطوير برنامج باستخدام تعليمتى ON/GOSUB و RETURN :

مثال (٨ - ٥) :

شركة تأجير سيارات لها ثلاثة مراكز في المملكة ، الأول في الرياض ، والثاني بجدة ، والثالث بالدمام ، سياستها في التأجير أن تسمح للمستأجر أن يقطع يومياً مسافة ١٥٠ كم بالسيارة مقابل مبلغ ١٥٠ ريالاً ، وأن يدفع ٥,٥٠ ريالاً عن كل كم يزيد على ذلك .

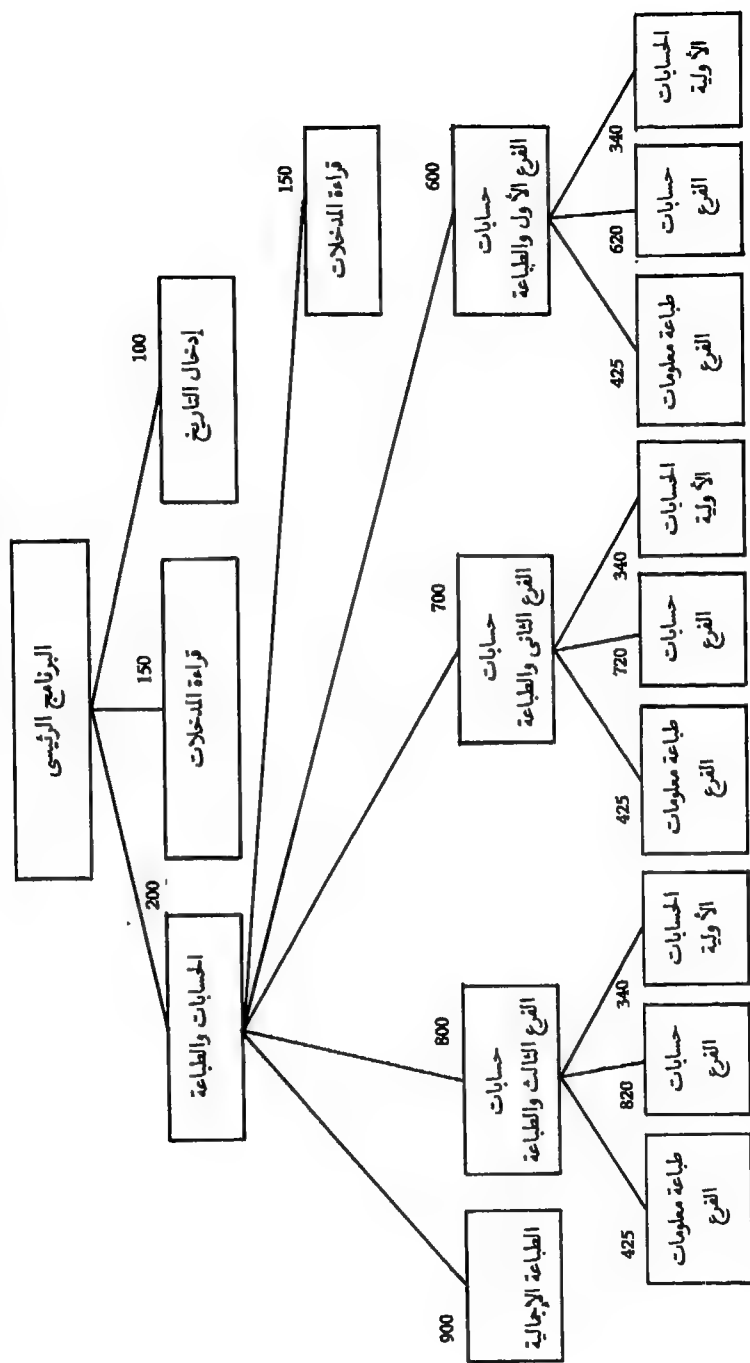
الهدف : ترغب الشركة في إيجاد المبلغ الإجمالي للدخل اليومي في كل من المراكز الثلاثة ، والدخل الإجمالي للشركة .

أولاً - خطوات الحل :

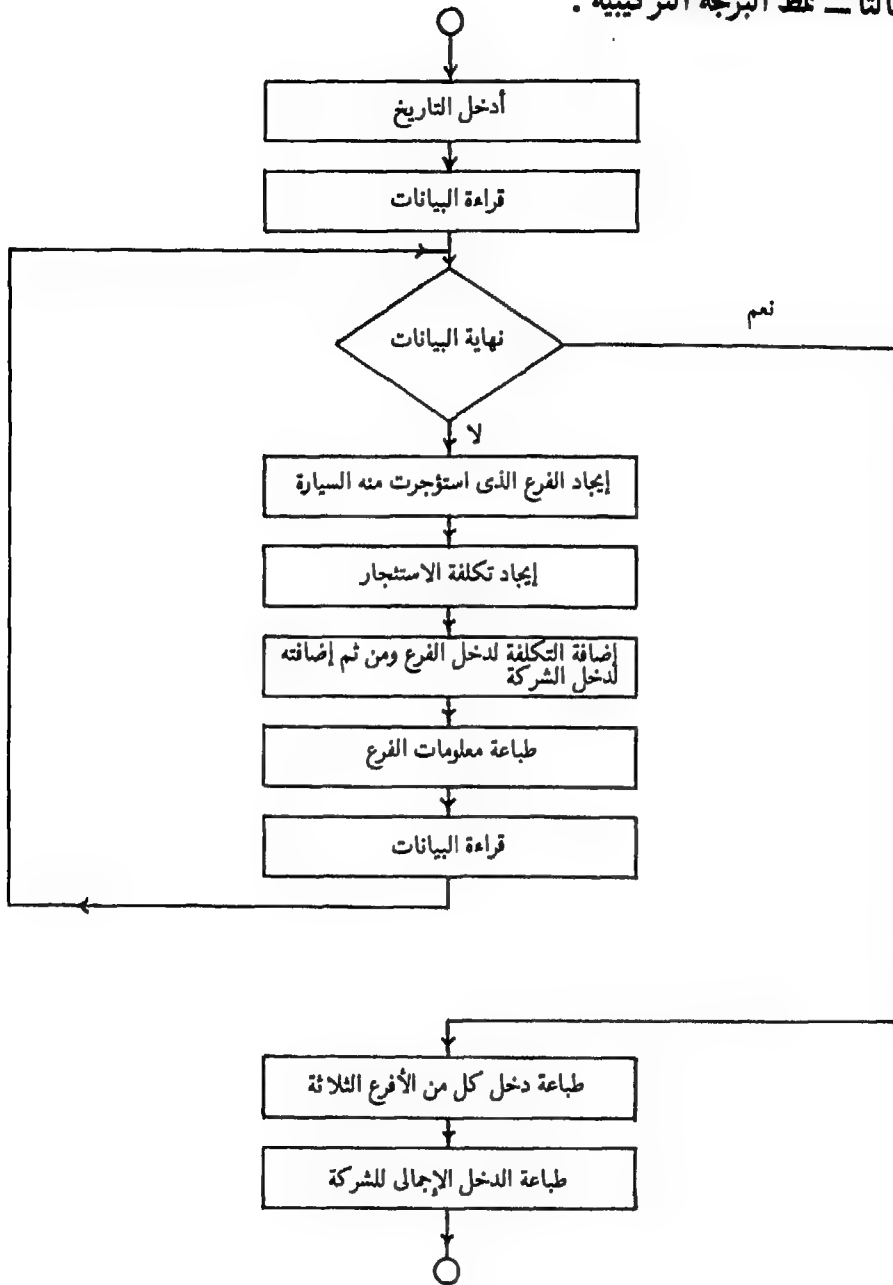
- ١ - الحصول على رمز المدينة ، رقم لوحة السيارة ، اسم المستأجر ، عداد السيارة وقت الاستئجار ، عداد السيارة عند الإعادة ، عدد أيام الاستئجار .
- ٢ - إيجاد عدد الكيلومترات المستخدمة .
- ٣ - إيجاد عدد الكيلومترات الزائدة .
- ٤ - إيجاد التكلفة الرئيسية والزائدة .
- ٥ - إيجاد الدخل لكل فرع وللشركة ككل .
- ٦ - طباعة اسم المدينة ، رقم لوحة السيارة ، اسم المستأجر ، عداد السيارة وقت الاستئجار ، عداد السيارة عند الإعادة ، عدد الكيلومترات المصرح بها ، عدد الكيلومترات الزائدة ، التكلفة الرئيسية ، التكلفة الزائدة ، التكلفة الإجمالية لكل فرع ، ومجموع التكلفة الكلية .

ثانياً: يوضح الشكل التالى الهيكل الهرمى ذا المستويات الثلاثة.

— الهيكل الهرمى مع أرقام عبارات البرامج فى الشكل (٨-٥) :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (٨ - ٥)

برنامج لحساب الدخل الإجمالي لمؤسسة تأجير سيارات والدخل التفصيلي لأفرعها

```

10 REM استدعاء برنامج فرعي
20 GOSUB 100
30 REM استدعاء برامج فرعية
40 GOSUB 300
45 GOSUB 800
47 GOSUB 1000
50 END
100 REM برنامج فرعي للمحول على التاريخ وطباعة المناويع
120 INPUT "ادخل التاريخ اليوم الشهر السنة كالتالي";D,M,Y
140 PRINT "=====
150 PRINT TAB(20); " شركة ا ب ج لتأجير السيارات";TAB(60);Y;"/";M;"/";D
155 PRINT "=====
160 PRINT " رقم اسم المدة عدد كم كم تكلفة تكلفة
170 PRINT " النوع الزبون استأجر اعادة الايام مستخدم راغد راغد عليه "
190 PRINT "=====
200 RETURN
300 REM برنامج فرعي للمحول على البيانات
310 REM
320 READ N,P#,N#,M1,M2,D
325 REM للاستفسار عن نهاية البيانات
327 IF N = 0 THEN RETURN
330 REM للمحول على عدد الكيلومترات المستخدمة
340 LET M = M2 - M1
350 REM للايجاد عدد الكيلومترات المسموحة
360 LET A = D * 150
370 REM للايجاد عدد الكيلومترات الزائدة
375 LET K = M - A
380 REM للايجاد تكلفة الكيلومترات الزائدة
385 LET E = K * 3.5
388 REM للايجاد التكلفة الكلية
390 LET C = E + D * 100
420 DN N GOSUB 800,700,800
425 PRINT TAB(1);C;TAB(10);E;TAB(19);K;TAB(27);M1;TAB(34);D;TAB(37);M2;TAB(44);M1;
TAB(52);N;TAB(65);P#;C#
430 REM
440 GOTO 320
450 REM البيانات
500 DATA 1,24581,ALI AHMED,25430,28340,7
510 DATA 2,31890,HASAN SALIH,20710,34580,2
520 DATA 3,72098,RADI AHMED,45430,48340,5
530 DATA 2,90811,ALI HASAN,55430,58040,9
540 DATA 1,56901,ANAS ALI,44587,48098,8
550 DATA 3,10981,SAMEER RADI,8123,12460,8
580 DATA 0,00000,AAAAAAAAA,00000,00000,0
800 REM برنامج فرعي للايجاد دخل فرع الرياض و اضافته على الدخل الكلي
820 LET C# = " الرياض "
830 LET R1 = R1 + C
840 T = T + C
850 RETURN
700 REM برنامج فرعي للايجاد دخل فرع الدمام و اضافته على الدخل الكلي
710 LET C# = " الدمام "
720 LET R2 = R2 + C
730 LET T = T + C
740 RETURN

```

```

800 REM برنامج فرعي للايجاد دخل فرع جده و اضافته على الدخل الكلي
810 LET C% = "جده"
820 LET R3 = R3 + C
830 LET T = T + C
840 RETURN
900 REM برنامج فرعي لطباعة الدخل الاجمالي لكل فرع
910 PRINT "=====
=====
920 PRINT TAB(20);R1;TAB(30);" = اجمالي دخل فرع الرياض"
930 PRINT TAB(20);R2;TAB(30);" = اجمالي دخل فرع الدمام"
940 PRINT TAB(20);R3;TAB(30);" = اجمالي دخل فرع جده"
945 PRINT "=====
=====
950 RETURN
1000 REM برنامج فرعي للايجاد الدخل الاجمالي للشركة و طباعته
1020 LET R = R1 + R2 + R3
1030 PRINT TAB(20);R;TAB(30);" = اجمالي الدخل للشركة"
1035 PRINT "=====
=====
1040 RETURN
1105 PRINT

```

وفي حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

شركة ا ب ج لتأجير السيارات										1987 / 12 / 30	
اسم المدينة	رقم اللوحة	الزبون	الاسم	عدد عند	عدد مستخدم	كم رأشه	تكاليف	تكاليف	كلية		
الرياض	24561	ALI AHMED	25430	28340	7	2810	1880	8510	7210		
الدمام	31890	HASAN SALIM	20710	34580	2	13850	13550	47425	47825		
جده	72098	RADI AHMED	45430	48340	5	2810	2160	7660	8080		
الدمام	9081	ALI HASAN	55430	58040	9	2810	1280	4410	5310		
الرياض	56901	ANAS ALI	44587	46098	6	1531	831	2208,5	2808,5		
جده	10981	SAMEER RADI	8123	12450	8	4327	3127	10844,5	11744,5		
=====											
						10018.5	=	اجمالي دخل فرع الرياض			
						52935	=	اجمالي دخل فرع الدمام			
						19804.5	=	اجمالي دخل فرع جده			
=====											
						82758	=	اجمالي الدخل للشركة			
=====											

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعليمات .

٢ - بين الأخطاء إن وجدت في كل من العبارات التالية :

10	IF A = 5 THEN B = 2	أ
10	IF A = 4 THEN 70 ELSE B = 12	ب
10	IF A = 4 AND 7 THEN 80	ج
10	IF A > A THEN B = A	د
10	IF B < 10 THEN ELSE C = B	هـ

٣ - بين نتائج تنفيذ كل من البرامج التالية :

10	READ A , B	أ -
20	IF A > B THEN C = A ELSE C = B	
30	PRINT C , A , B	
40	DATA 10 , 7 , 2 , 4	
50	END	
10	LET I = 0	ب -
20	READ A	
30	IF I > 5 THEN 90	
40	IF A > 0 AND A < 10 THEN I = I + 1	
50	PRINT I , A	
60	READ A	

70 COTO 20
80 DATA 10 , 20 , 7 , 2 , 1 , 3 , 4 , 17
90 END

٤ - بين القيم التي يجب أن يحتويها المتغير A لكي يتم تنفيذ العبارات التالية بدون أخطاء .

10	ON A	COTO	10 , 20 , 30 , 40	أ
10	ON A - 10	COTO	70 , 80 , 90 , 100 , 90	ب
10	ON ((A / 100) - 1)	COTO	60 , 50 , 70 , 90	ج
10	ON A * .5	COSUB	100 , 200 , 300	د

٥ - اكتب برنامجاً يعمل على قراءة بيانات أربعة أصناف من البضائع ، مع العلم بأن بيانات الصنف مكونة من (رقم الصنف ، عدد القطع المباعة ، سعر كل قطعة) ومن ثم إيجاد الثمن الإجمالي لمبيعات كل صنف .

أ - باستخدام تعليمة IF / THEN
ب - باستخدام تعليمة ON / GOTO

٦ - محل السلمان لبيع أدوات وملحقات التصوير يمنح خصماً لزملائه الجملة كالتالي :

تكلفة البضائع المشتراة	الخصم %
١ - أقل من ٢٠٠ ريال	صفر
٢ - من ٢٠٠ إلى أقل من ٥٠٠ ريال	٥
٣ - من ٥٠٠ إلى أقل من ٨٠٠ ريال	٧
٤ - ٨٠٠ ريال فأكثر	١٠

طور برنامجاً لعمل التالى :

١ . أدخل أرقام القطع وأسعارها وكمياتها لكل زبون .

٢ . احسب المجموع .

٣ . احسب الخصم .

٤ . احسب صافى المبلغ المستحق من زبون الجملة .

٧ — محل لبيع الأسماك بالجملة يستقبل طلبيات من أنواع ثلاثة من الزبائن :

— مطاعم .

— مجتمعات سكنية .

— خاص من أفراد .

ومن الممكن أن تحتوى الطلبية على الأصناف التالية : السمك ، الخبز ،
والحلويات . وتوجد تسعيرة خاصة لكل نوع من الزبائن مع خصم من مجموع قيمة
الطلبية كالتالى :

١٠ ٪ للمطاعم فوق ٢٠٠ ريال ، ٧ ٪ للسكنى فوق ١٥٠ ريالاً ، ٣ ٪ للأفراد فوق
٣٠ ريالاً .

طور برنامجاً لحساب قيمة الفاتورة لكل طلبية ، وإيجاد مجموع المبيعات لكل نوع من
الزبائن والمجموع الكلى للمبيعات فى نهاية كل يوم .

٨ — يقوم مكتب محامين بتحصيل الديون المتأخرة للشركات ، ويقوم المكتب

بحساب العمولة كنسبة من المبلغ المحصل ، وذلك حسب الجدول التالى :

المبلغ المحصل بالريال	العمولة ٪
أقل من ٢٠٠٠	٣
٢٠٠٠ إلى أقل من ٤٠٠٠	٧
٤٠٠٠ إلى أقل من ٧٠٠٠	١٠
٧٠٠٠ فأكثر	١٥

٩ - يؤجر محل (س ص ع) أشرطة الفيديو. ويمثل الجدول التالى أجر اليوم لأنواع الأشرطة المختلفة .

نوع الشريط	الأجر لليوم (بالريال)
١ - محاضرات	١٥
٢ - أفلام كرتون	٢٠
٣ - أفلام عادية للأطفال	٣٠
٤ - أفلام عادية للكبار	٣٥

يود صاحب المحل الحصول على مجموع الإيرادات لكل نوع من الأشرطة والمجموع الكلى للإيرادات ، إذا علم عدد الأشرطة المؤجرة وعدد الأيام .

١٠ - يحتوى أحد المجمعات السكنية على أربعة أنواع من الشقق ، ولكل نوع منها إيجار شهري ، كما هو مبين فى الجدول التالى :

نوع الشقة	الإيجار (بالريال)
شقة أ غرفة نوم واحدة	٨٠٠
شقة ب غرفتا نوم	١٠٠٠
شقة جـ ٣ غرف نوم	١٢٥٠
شقة د ٤ غرف نوم	١٥٠٠

بالإضافة إلى الإيجار يقوم صاحب المجمع بتوزيع مصاريف الإنارة الخارجية والمياه بنسبة ١ : ٢ : ٣ : ٤ وذلك حسب نوع الشقة .
طور برنامجاً لحساب مجموع الإيجار الشهري للشقة الواحدة .

استخدام الدالات فى العمليات الحسابية

فى كثير من التطبيقات ، نجد أن أكثر من برنامج يريد القيام بنفس المعالجة أو أن البرنامج الواحد يكرر تنفيذ نفس المعالجة . وقد أدى هذا إلى استخدام البرامج الفرعية كما أشير إليه فى الأمثلة السابقة ، حيث يتم تنفيذ التعليمات المبينة فى البرنامج الفرعى بواسطة تعليمة COSUB .

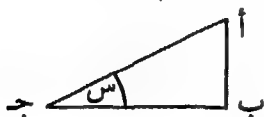
ومن هذه العمليات الحسابية المتكررة الاستخدام ما يمكن تمثيله بمعادلة جبرية أو أكثر . وهذه المعادلة قد تكون عامة ، يمكن استخدامها فى كل التطبيقات ، أو خاصة بتطبيق معين ، ومثال النوع الأول حساب جيب زاوية معينة ، والنوع الثانى تحويل درجة حرارة من مئوية إلى فهرنهايتية . وقد احتوت لغة بيسك تعليمات خاصة للتعامل مع المعادلات من هذا النوع ، وذلك عن طريق استخدام أسماء الدالات التى ستقوم بالعمليات الحسابية . وفى النوع الأول دالات مبرمجة ومخزنة فى الحاسب وذات أسماء تحدد طبيعة العملية الحسابية التى ستقوم بها ، مثل : جيب ، القيمة المطلقة ... إلخ .

أما فى النوع الثانى فتسمح لغة بيسك للمبرمج بأن يحدد اسماً للدالات والمتغيرات المتضمنة بها والعلاقة الجبرية فيما بينها . ويتم استخدام عبارات الدالات هذه فى البرنامج بحرية تامة من قبل المبرمج وفى أى موضع يشاء ، وستحدث فى هذا الفصل عن كيفية استخدام هذه الدالات .

Built - in الدالات المبرجة / المبنية

مقدمة عن الدالات المبرمجة :

إذا أردنا إيجاد جيب الزاوية (أ ج ب) SINE في المثلث القائم الزاوية ، نجد



أنها تساوى
$$\frac{\text{طول الضلع أ ب}}{\text{طول الضلع أ ج}} =$$

هذا في حالة معرفة أطوال أضلاع المثلث .

أما في حالة عدم معرفة أطوال أضلاع المثلث ومعرفة الزاوية (س) نجد أن جيب

الزاوية (س) =

$$\begin{aligned} & \left(\frac{س^{11}}{مفكوك^{11}} - \frac{س^9}{مفكوك^9} \right) + \left(\frac{س^7}{مفكوك^7} + \frac{س^5}{مفكوك^5} + \frac{س^3}{مفكوك^3} \right) - س \\ & \dots\dots\dots (\dots\dots) + \left(\frac{س^{15}}{مفكوك^{15}} - \frac{س^{13}}{مفكوك^{13}} \right) + \end{aligned}$$

ويتضمن إيجاد جيب الزاوية θ حسب المعادلة السابقة، الخطوات التالية :

١- إيجاد قيم البسط حيث إن قيمة س مرفوعة إلى قوة الأرقام الفردية ٣ ، ٥

٢- إيجاد مفكوك كل من هذه الأرقام الفردية (أى إيجاد عناصر كل رقم منها وضربها بعضها فى بعض) ومن ثم إيجاد الناتج .

٣- قسمة ناتج الخطوة الأولى على ناتج الخطوة الثانية وإضافة خارج القسمة إلى الناتج الكلي .

٤- تكرار الخطوات ١-٣ بعدد المرات اللازمة لتحقيق الدقة المطلوبة في الإجابة .

٥- طرح الناتج الكلي من قيمة الزاوية s حيث إن ناتج الطرح يساوى جيب الزاوية .

فلو أراد عدة مبرمجين الحصول على جيب الزاوية ، كل في تطبيقه الخاص به ، لا يضطر كل واحد منهم لكتابة برنامج فرعى بشكل خاص ومستقل عن الآخر ، وكذلك يمكن لأى منهم أن يستعير البرنامج الفرعى من برنامج آخر وطباعته فى برنامجه ، أو أن تخزن الجداول الخاصة بجيب الزاوية — وغيرها من حساب المثلثات — فى البرنامج وتسترجع عند الحاجة إليها .

وأى من هذه الإجراءات يبدو مستهلكاً للوقت وذافاعلية قليلة ، لذلك فهذه وغيرها من العمليات الشبيهة بها قد تمت برمجتها كما مر ذكره ، بحيث يتمكن المبرمج من استدعاء الدالة المطلوبة باستخدام الاسم الخاص بها فقط مصحوباً بالرقم المناسب لكل دالة ، ومقدار الزاوية فى هذه الحالة .

وتتبع قواعد الدالات الشكل العام التالى :

FUNCTION		الشكل العام لعبارة الدالة	
xxxxx	xxxx	xxx	(y y y)
↓	↓	↓	↓
رقم السطر	تعلية	اسم الدالة	متغير/قيمة/ عملية حسابية

ففى مثالنا للحصول على جيب الزاوية (SINE) نستخدم العبارة التالية :

10 PRINT SIN (S)

فإذا كانت قيمة S = 30 نجد أن القيمة المطلوبة ستكون 0.5

بعض الدالات المبرجة والمزودة فى معظم الحاسبات :

١ — للحصول على جيب الزاوية نستخدم الدالة SIN

مثال : جيب الزاوية 20 20 A = SIN LET 10

٢ — للحصول على جيب تمام الزاوية نستخدم الدالة COS

مثال : جيب تمام الزاوية 40
10 LET A = COS (40)

٣ — للحصول على ظل الزاوية نستخدم الدالة TAN

مثال : ظل الزاوية 40
10 LET A = TAN (40)

٤ — للحصول على الجذر التربيعي نستخدم الدالة SQR

مثال : أوجد الجذر التربيعي. نستخدم الدالة 70

الحل :
10 LET A = SQR (70)

٥ — للحصول على قيمة الزاوية في حالة معرفة ظلها نستخدم الدالة ANT

مثال : أوجد قيمة الزاوية التي ظلها = (واحداً)

الحل :
10 PRINT ATN (1)

وفي حالة تنفيذ هذه العبارة نجد أن القيمة المطبوعة = ٩٥

٦ — للحصول على اللوغاريتم الطبيعي نستخدم الدالة LOG

مثال : أوجد اللوغاريتم الطبيعي للقيمة 100

الحل :
10 LET A = LOG (100)

٧ — للحصول على القيم صحيحة بدون كسور نستخدم الدالة INT ، حيث إنها

تعطينا القيم الصحيحة وتهمل الكسور في حالة كون القيم موجبة ، أما في حالة كون القيم سالبة فإنها تعمل على تقريبها للعدد الصحيح .

مثال :
10 LET A = 15.94

20 LET B = -24.31


```
30 PRINT INT (A) , INT(B)
```

```
40 END
```

وفي حالة تنفيذ هذه العبارات نجد أن القيم المطبوعة هي :

15

-25

أى تم تقريب القيمة السالبة إلى العدد الصحيح ، وأهملت الكسور في حالة القيمة الموجبة .

٨ — للحصول على القيمة المطلقة ، أى تحويل القيمة من سالبة إلى موجبة ، نستخدم الدالة . ABS

مثال : أوجد القيمة المطلقة للقيم 15 . 94 , -24 . 31

```
10 LET A = 15 . 94
```

الحل :

```
20 LET B = 24 . 31
```

```
30 PRINT ABS (A) , ABS (B)
```

```
40 END
```

وفي حالة تنفيذ هذه العبارة نجد أن القيم المطبوعة هي :

15 . 94

24 . 31

٩ — للحصول على الدالة الأسية نستخدم الدالة EXP وهذا يعنى أن القيمة المراد إيجاد الدالة الأسية لها ستكون (القوة/الأس) للقيمة 2.718282.

مثال : أوجد الدالة الأسية للقيمة 2

```
10 LET A = 2
```

الحل :

```
20 PRINT EXP (A)
```

```
30 END
```

وفي حالة تنفيذ هذه العبارة ستكون النتيجة = 7.389057

١٠ - للحصول على النسبة التقريبية نستخدم الدالة PI

مثال : أوجد النسبة التقريبية

الحل : 10 PRINT PI

وفي حالة تنفيذ هذه العبارة ستكون النتيجة المطبوعة = 3.1415926

١١ - للحصول على إشارة القيمة نستخدم الدالة SGN ، ففي حالة كون النتيجة سالبة

ستكون الإشارة تساوى -1 ، وفي حالة كون النتيجة صفراً ستكون الإشارة = 0

أما في حالة كون النتيجة موجبة فستكون الإشارة = +1

مثال : أوجد إشارة كل من القيمة 112 و 0 و 30

الحل : 10 PRINT SGN (112) , SGN (0) , SGN (-30)

وفي حالة تنفيذ هذه العبارة ستكون النتائج المطبوعة كما يلي :

+1 0 -1

١٢ - للحصول على الباقي بعد إجراء العمليات الحسابية نستخدم الدالة MOD

مثال : أوجد باقى قسمة A على B , مع العلم بأن 5 = A , 2 = B

الحل : 10 LET A = 5

20 LET B = 2

30 PRINT A MOD B

40 END

وفي حالة تنفيذ هذه العبارة ستكون النتيجة المطبوعة = 0.5

١٣ - للحصول على الأرقام العشوائية (RANDOM NUMBERS) نستخدم الدالة

RND ، وتستخدم هذه الدالة في حالة عدم توفر البيانات الحقيقية المراد

استخدامها في المعالجة ؛ لذا يتم استخدام هذه الدالة لتزويدنا بقيم تقريبية .

والشكل العام لعبارة الدالة RND			
RND : أ	X X X	ب : (X) RND	X X X
اسم الدالة	رقم السطر	قيمة أو اسم الدالة	رقم السطر
		اسم متغير على أن تكون	
		قيمتها بدون كسر.	

وباستخدام الدالة RND سنحصل على أرقام تتراوح قيمتها ما بين :
(.000000 - 1.0) .

تطوير برامج باستخدام الدالات المبرمجة :

وبعد هذا العرض الإجمالي للدالات ، سنتعرض لتطوير برامج باستخدام البعض منها .

مثال (٩ - ١) :

الهدف : إيجاد عوامل (FACTORS) عدد معين ، مع العلم بأن عوامل أى عدد هى العناصر التى يتكون منها العدد بحيث تكون مساوية له أو أقل منه ، ويقبل قسمة العدد عليها بدون كسر (أى بدون بواقي) ، فمثلاً عوامل الرقم خمسة هى خمسة وواحد .

أولاً - خطوات الحل :

١ - الحصول على العدد .

٢ - إيجاد عنصر العدد الذى سيكون مساوياً (١ إلى قيمة العدد)

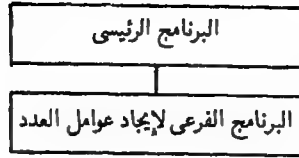
٣ - إيجاد ناتج قسمة العدد على العنصر .

٤ - الاستفسار عما إذا كان هذا العنصر عاملاً نعمل على طباعته ، وغير ذلك يهمل .

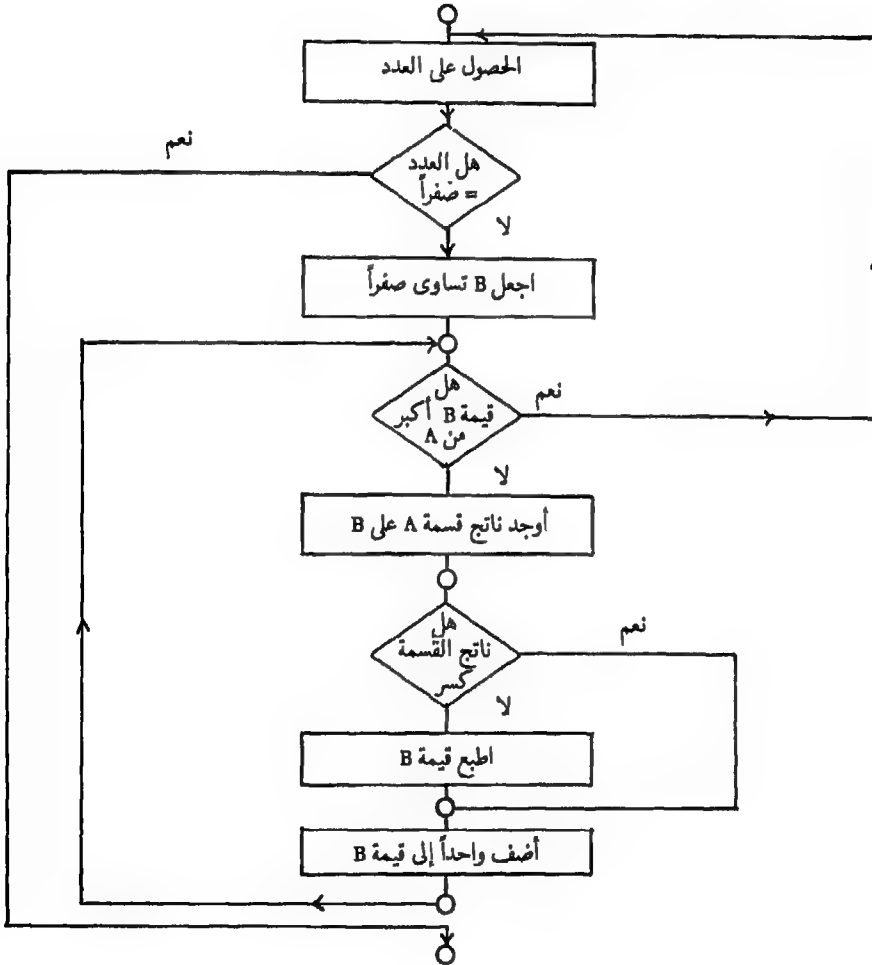
٥ - تكرار الخطوات ٢ - ٤ حتى نصل إلى قيمة العدد .

ويمكننا عمل ذلك باستخدام إحدى الدالتين INT أو MOD .

ثانياً - الهيكل الهرمي للبرنامج :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - أ . البرنامج في الشكل التالي ، باستخدام دالة INT :

شكل (٩ - ١)

برنامج لإيجاد عوامل عدد باستخدام الدالة INT	
10	REM استدعاء برنامج فرعي
20	GOSUB 40
30	END
40	REM برنامج فرعي لإيجاد عوامل عدد
50	INPUT "ادخل العدد المراد إيجاد عوامله أو صفر 0 للتوقف";A
60	IF A = 0 THEN 200
70	REM أستاذ القيمة الأولية للمتغير B الذي سيحتوي على عناصر العدد
80	LET B = 0
90	REM ستفسر عن قيمة العنصر إذا كانت أكبر من قيمة العدد
100	IF B > A THEN 180
110	REM إيجاد ناتج لقسمه العدد على العنصر بدون كسور
120	LET C = INT(A/B)
130	REM ستفسر إذا كان العنصر عاملاً
140	IF (C*B) <> A THEN 180
145	PRINT TAB(10);A;TAB(13);" " ;TAB(33);B;TAB(39);" الرقم "
150	REM إضافة واحد لمتغير العناصر و الموده للإستفسار
160	LET B = B + 1
170	GOTO 100
180	PRINT "-----"
190	GOTO 50
200	RETURN

ب - تعديل البرنامج السابق لإيجاد عوامل العدد باستخدام دالة MOD حيث
نعمل على تغيير كل من العبارات التالية :

(١) عبارة رقم 120

120 LET C = A MOD B

وهنا تتم قسمة قيمة المتغير A على قيمة المتغير B وسيحتوى المتغير C على باقى
القسمة (الكسور) . ومن ثم نستفسر عن قيمة المتغير C . فإذا كانت القيمة لا تساوى
صفرأ (0) فذلك يعنى أن قيمة المتغير B لا تعتبر عاملاً من عوامل العدد . ويتم ذلك
بتغيير العبارة رقم (140) إلى :

140 IF C < > 0 THEN 180

وبعد تنفيذ البرنامج حسب الطريقتين ، سنحصل على نفس النتائج كالتالى :

الرقم	1	أحد عوامل العدد	6
الرقم	2	أحد عوامل العدد	6
الرقم	3	أحد عوامل العدد	6
الرقم	6	أحد عوامل العدد	6

الرقم	1	أحد عوامل العدد	7
الرقم	7	أحد عوامل العدد	7

الرقم	1	أحد عوامل العدد	9
الرقم	3	أحد عوامل العدد	9
الرقم	9	أحد عوامل العدد	9

مثال (٩ - ٢) :

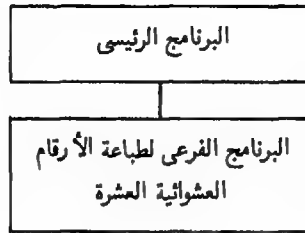
المهدف : الحصول على عشرة أرقام عشوائية * .

أولاً - خطوات الحل :

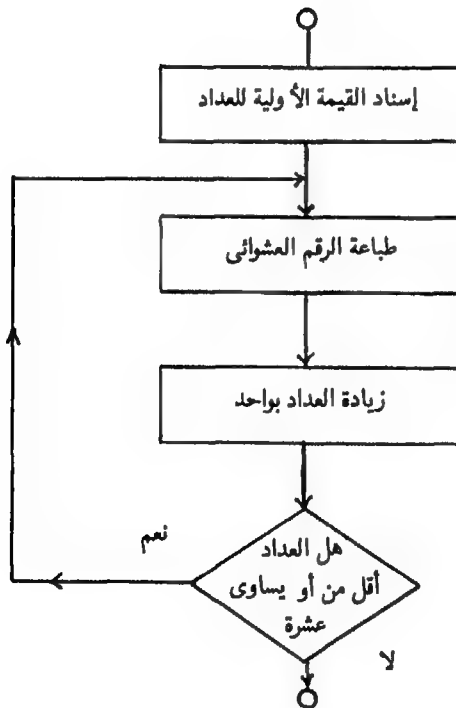
- ١ - إسناد القيمة الأولية 1 لعداد الأرقام العشوائية .
- ٢ - طباعة الرقم العشوائى .
- ٣ - زيادة العداد بواحد والاستفسار عن عدد الأرقام .
- ٤ - إذا كان عدد الأرقام المطبوعة أقل من عشرة نكرر الخطوات ٢ - ٣ وإلا نتوقف .

* إن اختيار رقم عشوائى من بين عدة أرقام ، يعنى أن أى رقم منها له نفس الاحتمال أو الفرصة في أن يتم اختياره .

ثانياً - الهيكل الهرمي للبرنامج :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (٩ - ٢)

برنامج للحصول على عشرة أرقام عشوائية باستخدام الدالة RND	
10 REM	استدعاء برنامج فرعي
20 GOSUB 40	
30 END	
40 REM	برنامج فرعي للحصول على عشرة أرقام عشوائية
50 REM	استناد القيمة الأولية (١) لعدد الأرقام
60 LET I = 1	
70 REM	طباعة الرقم العشوائي
80 PRINT RND	
90 REM	زيادة عدد الأرقام بواحد
100 LET I = I + 1	
110 REM	للاستفسار عن عدد الأرقام إذا كان أقل أو يساوي عشرة يذهب للطباعة
120 IF I <= 10 THEN 80	
130 REM	العودة للبرنامج الرئيسي
140 RETURN	

وفي حالة تنفيذ البرنامج ، ستظهر النتائج كالتالي :

.1213501
.651861
.8688612
.7297625
.7988531
7.369809E-02
.4903128
.454519
.1072496
.9505102

ولو أردنا الحصول على عشرة أرقام عشوائية أخرى ، وقمنا بتنفيذ البرنامج شكل (٩ - ٢) ، لوجدنا أن النتائج (القيمة التي ستطبع) ستكون نفس القيم التي حصلنا عليها سابقاً . وللحيلولة دون ذلك نستخدم الدالة RANDOMIZE في بداية البرنامج كما هو مبين في الشكل التالي :

RANDOMIZE	الشكل العام لعبارة الدالة
<p style="text-align: center;">x x x</p> <p style="text-align: center;">رقم السطر</p>	<p style="text-align: center;">RANDOMIZE</p> <p style="text-align: center;">اسم الدالة</p>

• جهاز IBM يتطلب طباعة رقم بين 32768 إلى 32767
بعد اسم الدالة أو عند التنفيذ

فإذا أضفنا العبارة التالية في بداية البرنامج الفرعى في شكل (٩ - ١) ، نجد أنه في كل مرة يتم بها تنفيذ البرنامج ، سنحصل على عشرة أرقام عشوائية ، تختلف عن الأرقام التي حصلنا عليها في المرة السابقة .

نلاحظ أن جميع الأرقام العشوائية التي حصلنا عليها حتى الآن كانت عبارة عن كسور تتراوح قيمها ما بين (0 و 1) ، ويمكن الحصول على رقم عشوائى صحيح (بدون كسر) يقع ما بين 1 ، 10 ، وذلك باستخدام العبارة التالية :

10 LET A = INT (10 * RND + 1)

وسيبين البرنامج التالى كيفية الحصول على أرقام عشوائية صحيحة ذات مدى محدد .

• لا داعى هنا للخوض في كيفية عمل هذه التعليمة ؛ إذ أن لها معانى إحصائية تؤثر في الخطوات الموصلة إلى الرقم العشوائى .

مثال (٩ - ٣) :

الهدف : الحصول على ثلاث مجموعات من الأرقام العشوائية الصحيحة التى تقع ما بين :

أ — 10 , 1

ب — 75 , 15

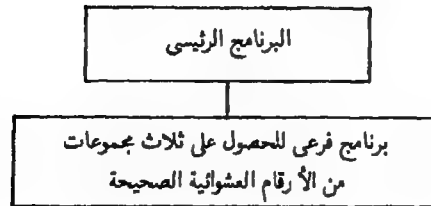
ج — 95 , 80

وكل مجموعة تحتوى على الأرقام العشوائية .

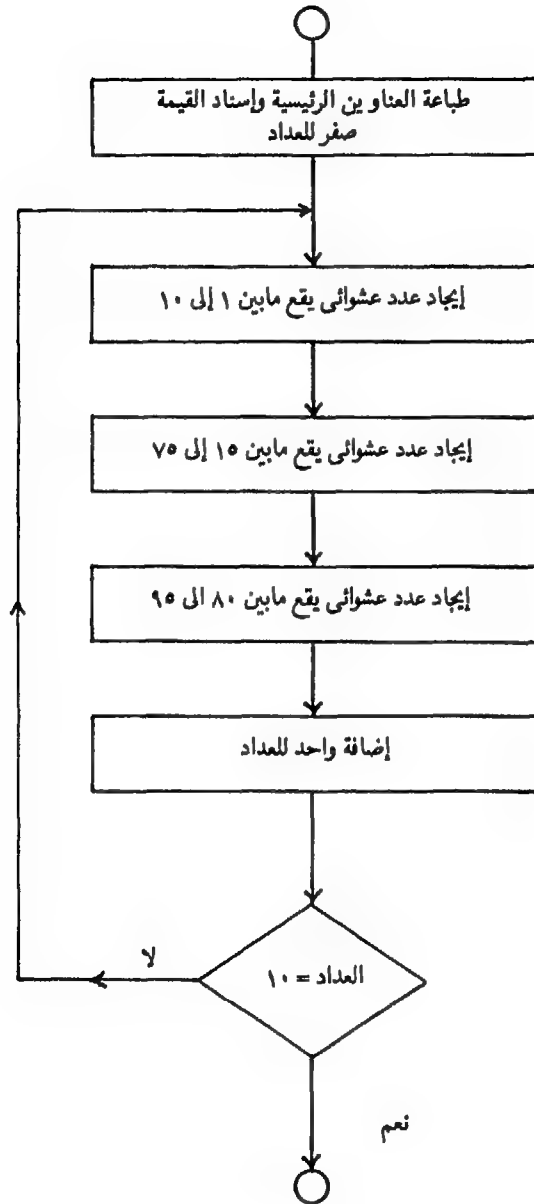
أولاً — خطوات الحل :

- ١ — إسناد القيمة الأولية (1) لعداد الأرقام العشوائية .
- ٢ — الحصول على الرقم العشوائى فى المجموعة الأولى .
- ٣ — الحصول على الرقم العشوائى فى المجموعة الثانية .
- ٤ — الحصول على الرقم العشوائى فى المجموعة الثالثة .
- ٥ — طباعة الأرقام الثلاثة وزيادة العداد بواحد .
- ٦ — الاستفسار عن العداد، إذا كان أقل من ١١ نكرر الخطوات من (٢ - ٥) ولا نتوقف .

ثانياً — الهيكل الهرمى للبرنامج :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (٩-٣)

برنامج للحصول على ثلاث مجموعات من الأرقام العشوائية الصحيحة		
10	REM	استدعاء برنامج فرعي
20	GOSUB	40
30	END	
40	REM	برنامج فرعي للحصول على ٣ مجموعات من الأرقام
50	REM	طباعة المناوين الرئيسية
60	PRINT	TAB(30); "ثلاثة مجموعات من الأرقام العشوائية"
70	PRINT	TAB(10); "95-80"; TAB(20); "75-15"; TAB(30); "10-1"
80	PRINT	TAB(10); "-----"; TAB(20); "-----"; TAB(30); "-----"
90	REM	إسناد القيمة الأولى للعداد
100	LET	I = 1
110	LET	N = RND
120	REM	10-1 للحصول على قيمة تقع ما بين
130	LET	A = INT(N*10)
140	REM	75-15 للحصول على قيمة تقع ما بين
150	LET	N = RND
160	IF	N < 15/75 THEN 150 : REM
170	LET	B = INT(75*N)
180	REM	99-80 للحصول على قيمة تقع ما بين
190	LET	N = RND
200	IF	N < 80 / 95 THEN 190 : REM
210	LET	C = INT(95*N)
220	REM	طباعة الأرقام الثلاثة التي تم الحصول عليها
230	PRINT	
240	PRINT	TAB(10); C; TAB(20); B; TAB(30); A
250	LET	I = I + 1
260	REM	للاستفسار عن نهاية عداد الأرقام
270	IF	I <= 10 THEN 110
280	RETURN	

وفي حالة تنفيذ البرنامج ، تظهر النتائج كالتالي :

ثلاثة مجموعات من الأرقام العشوائية		
95-80	75-15	10-1
82	48	1
90	59	7
92	39	7
88	71	3
94	42	5
89	49	2
89	66	3
88	30	3
88	38	2
86	41	6

مثال (٩ - ٤) :

يصل الزبائن لمحطة بنزين بطريقة عشوائية وبمعدل واحد كل خمس دقائق

$$(٥ \times ٦٠ = ٣٠٠ \text{ ثانية}) .$$

الهدف : استخدام نظام التظاهر Simulation للحصول على التالى :

١ - إيجاد عدد الزبائن الذين سيصلون للمحطة فى خلال ساعة ما .

٢ - إيجاد عدد المرات التى سيكون بها موظف المحطة مشغولاً .

٣ - إيجاد عدد الزبائن الذين انتظروا قبل أن تتم خدمتهم .

ملاحظة : سيتم الحصول على رقم عشوائى فى كل دقيقة (٦٠ ثانية) . وبما أن وصول الزبائن هو بمعدل واحد كل خمس دقائق للمحطة فإنه :

أ - عدد الأرقام العشوائية التى سيتم الحصول عليها يساوى $\frac{٣٠٠}{٦٠} = ٥$

ب - يمكن استخدام القيمة من (صفر إلى $\frac{٦٠}{٣٠٠}$ أو ٢) كمؤشر لوصول زبون للمحطة .

أولاً - خطوات الحل :

١ - الحصول على الرقم العشوائى .

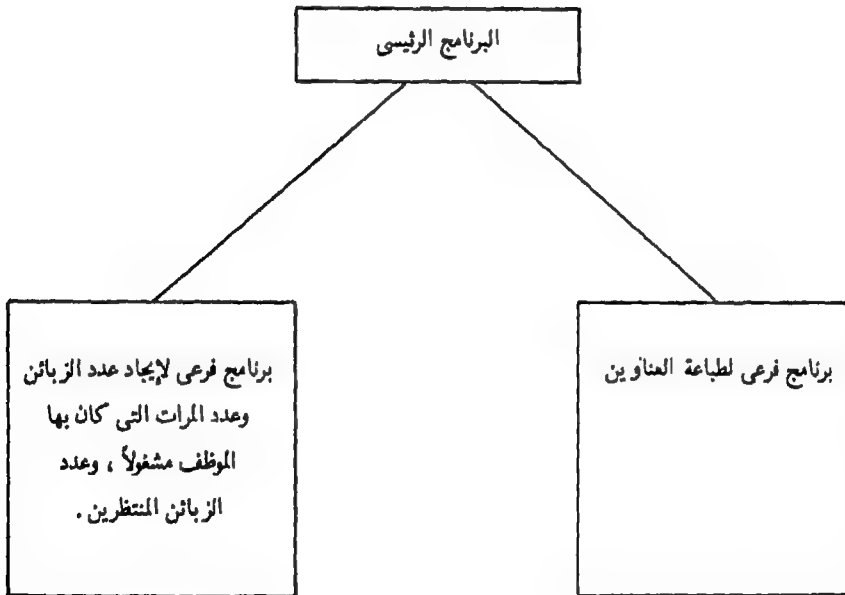
٢ - الاستفسار عن وصول زبون للمحطة ، وذلك بمقارنة قيمة الرقم العشوائى بالقيمة ٠,٢ .

٣ - الاستفسار عن نهاية الخدمة للزبون .

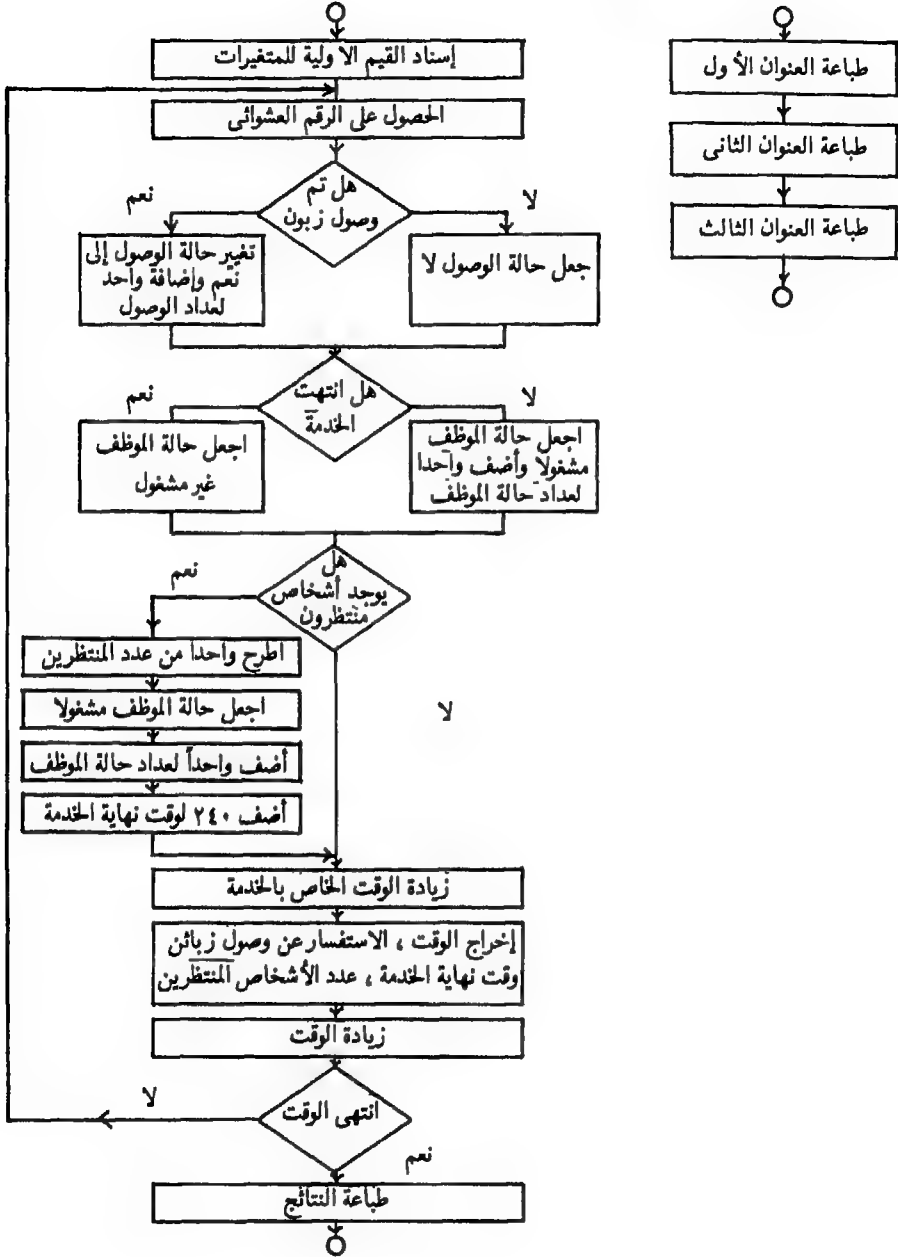
٤ - الاستفسار عن عدد الأشخاص المنتظرين للخدمة .

- ٥ - إخراج الوقت ، الاستفسار عن وصول زبون ، الحالة التي بها الموظف وقت نهاية الخدمة ، عدد الأشخاص المنتظرين .
- ٦ - تكرار الخطوات من (١ - ٥) إلى أن تنتهي الفترة الزمنية المرادة (٦٠ دقيقة) .
- ٧ - إخراج عدد الزبائن الذين وصلوا للمحطة ، عدد المرات التي كان بها الموظف مشغولاً ، عدد الزبائن الذين انتظروا .

ثانياً - الهيكل الهرمي للبرنامج :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (٩ - ٤)

برنامج خاص بعمليات انتظار الزبائن أمام محطات البنزين

```

10 REM      استدعاء برنامج فرعي
20 GOSUB 50
30 REM      استدعاء برنامج فرعي
35 GOSUB 100
40 END
50 REM      برنامج فرعي لطباعة المتأولين الرئيسي
60 PRINT TAB(20); " 1 محطة بنزين رقم "
70 PRINT "عدد المنتظرين  نهاية الخدمة  حالة الموقف  هل وصل زبون  الوقت "
80 PRINT "-----"
90 RETURN
100 REM     برنامج فرعي للاستخدام نظام التظاهر
110 REM     اسناد القيم الا وليه للمتغيرات
120 LET W = 0
130 LET S = 0
140 LET A$ = " "
150 LET E$ = "مكفول"
155 LET P$ = "###      ###.##      \      \      \      \      ### +
160 LET T = 0
170 REM     الحصول على الرقم العشوائي
180 LET R = RND
190 REM     الا ستفسار عن وصول زبون
200 IF R > .2 THEN 245
210 REM     اضافة واحد لمداد المنتظرين
220 LET W = W + 1
230 LET A$ = "نعم"
240 REM     الا ستفسار عن عدد المنتظرين
242 GOTO 250
245 REM     تفجير حالة الوصول الى لا
247 LET A$ = "لا"
250 IF T >= S THEN 280
260 REM     لطباعة النتائج و الحاله
270 GOTO 380
280 REM     اسناد فراغ لحاله الموقف
290 LET E$ = " "
300 REM     الا ستفسار عن عدد المنتظرين
310 IF W < 1 THEN 360
320 REM     طرح واحد من عدد المنتظرين
330 LET W = W - 1
340 LET E$ = "مكفول"
350 LET S = T + 240
380 PRINT TAB(8);W;TAB(19);S;TAB(34);E$;TAB(51);A$;TAB(67);T
370 LET T = T + 80
380 REM     للا ستفسار عن نهاية البحث و التظاهر
390 IF T <= 1200 THEN 180
395 PRINT "=====
=====
400 RETURN

```


وفي حالة تنفيذ البرنامج السابق ، منحصل على النتائج التالية :

محطة بنزين رقم 1

الوقت	هل وصل ذبون	حالة الموظف	نهاية الخدمة	عدد المنتظرين
0	نعم	مشغول	240	0
60	لا	مشغول	240	0
120	لا	مشغول	240	0
180	لا	مشغول	240	0
240	لا		480	0
300	نعم		480	1
360	لا		480	1
420	لا		480	1
480	نعم	مشغول	720	1
540	لا	مشغول	720	1
600	لا	مشغول	720	1
660	لا	مشغول	720	1
720	لا	مشغول	960	0
780	لا	مشغول	960	0
840	لا	مشغول	960	0
900	لا	مشغول	960	0
960	لا		1200	0
1020	لا		1200	0
1080	لا		1200	0
1140	لا		1200	0
1200	لا		1440	0

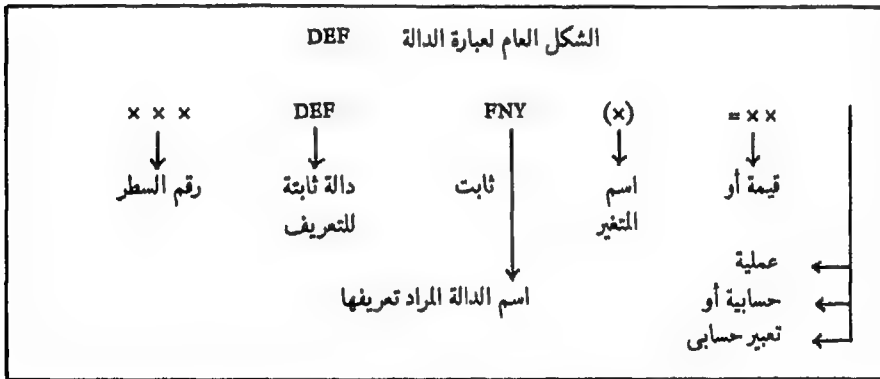
=====

الدالات المعرفة من قبل المبرمج

مقدمة عن الدالات المعرفة :

حتى الآن تم التعرض لبعض الدالات الجاهزة Built - in Functions والتي صممت وكتبت بواسطة الشركات المصنعة للأجهزة ، لكن توجد هناك حالات أو عمليات يرغب المستخدم في القيام بها ، في أكثر من خطوة في البرنامج ، أولاً أكثر من جزء من البيانات .

يمكننا عمل ذلك بتحديد نوع العمليات المراد إجراؤها باستخدام التعليمة DEF وهي عبارة عن دالة جاهزة. يمكننا تعريف العمليات المرادة وإسنادها إلى اسم الدالة ، ومن ثم استدعاؤها في أي جزء من أجزاء البرنامج ، كما هو الحال في تعليمة GOSUB وذلك حسب الشكل التالي :



حيث إن :

- DEF تستخدم كمؤشر للحاسب بأن ما يليها هو تعريف لدالة .
- FN اختصار لـ Function Name وتستخدم كمؤشر للحاسب بأن الحرف الذي يليها هو اسم الدالة المراد تعريفها .

Y — اسم الدالة، ويجب أن يكون أحد الحروف الألفبائية من A إلى Z .

(X) — اسم المتغير الذى سيحوى القيمة المراد معالجتها عند استدعاء الدالة المعرفة .

XX — يمكن أن يكون قيمة أو عملية حسابية (جمعاً ، ضرباً ، قسمة ...) أو تعبيراً حسابياً (معادلة جبرية) .

تطوير برنامج باستخدام الدالة المعرفة :

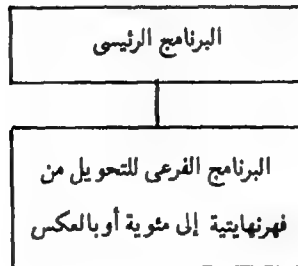
مثال : (٩ - ٥)

الهدف : باستخدام تعليمة الدالة DEF ، إيجاد ما يماثل الدرجات المثوية بالفهرنهايتية وبالعكس .

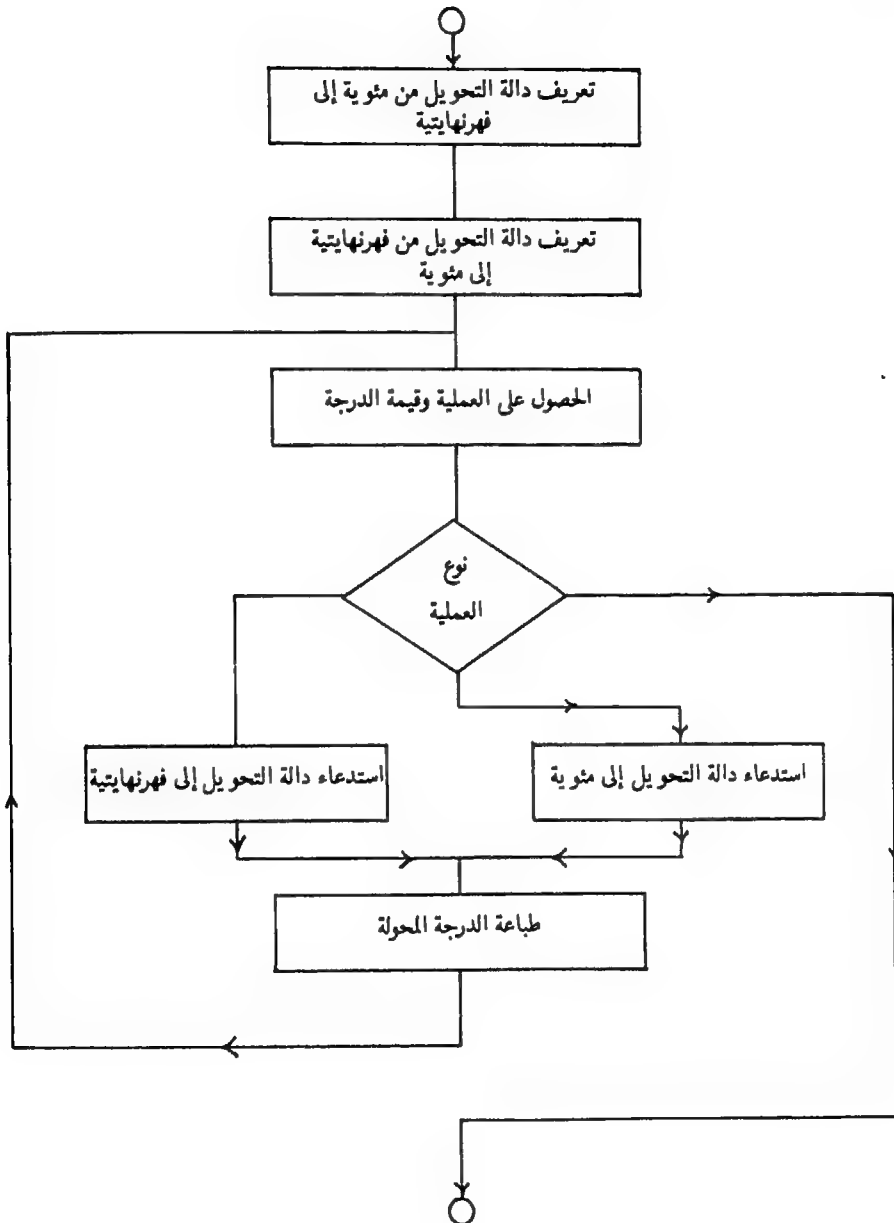
أولاً — خطوات الحل :

- ١ — تعريف دالة التحويل من المثوية إلى الفهرنهايتية .
- ٢ — تعريف دالة التحويل من الفهرنهايتية إلى المثوية .
- ٣ — الحصول على نوع العملية وقيمة الدرجة .
- ٤ — استدعاء الدالة حسب نوع العملية ، ومن ثم طباعة ما يماثلها .

ثانياً — الهيكل الهرمى للبرنامج



ثالثا - غط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (٩ - ٥)

برنامج لتحويل الدرجات المئوية إلى فهرنهايت وبالعكس باستخدام الدالة المعرفة	
10 REM	استدعاء برنامج فرعي
20 GOSUB 40	
30 END	
40 REM	برنامج فرعي لتحويل الدرجات المئوية إلى فهرنهايت وبالعكس
50 REM	تعريف دالة التحويل من مئوية إلى فهرنهايت
60 DEF FNF(T) = ((9/5) * T) + 32	
70 REM	تعريف دالة التحويل من فهرنهايت إلى مئوية
80 DEF FNC(T) = (5/9) * (T - 32)	
90 REM	الحصول على نوع العملية وقيمته الدرجة
100 PRINT TAB(35);	"لا شيء إلا اختيارات الرئيسية"
105 PRINT	
110 PRINT TAB(45);	"الرمز نوع العملية"
120 PRINT TAB(45);	"-----"
125 PRINT	
130 PRINT TAB(30);	1 "للتحويل من فهرنهايت إلى مئوية"
140 PRINT TAB(30);	2 "للتحويل من مئوية إلى فهرنهايت"
150 PRINT TAB(30);	3 "إنهاء العملية والتوقف"
160 PRINT TAB(30);	"-----"
165 PRINT	
170 INPUT	"N, D" أدخل رمز العملية المطلوبة والدرجة مفضولتان بفصله
180 REM	للاستفسار عن نوع العملية المطلوبة
190 ON N GOTO 250, 230, 270	
200 IF N >= 1 AND N < 3 THEN 100	
210 PRINT	"-----" خطأ في الإدخال
220 GOTO 170	
230 LPRINT	"D : درجة مئوية تساوي " ; FNF(D) ; "فهرنهايتية"
240 GOTO 100	
250 LPRINT	"D : درجة فهرنهايت تساوي " ; FNC(D) ; "مئوية"
260 GOTO 100	
270 RETURN	

وفي حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

أ - شاشة الاختيارات :

لائحة الاختيارات الرئيسية

الرمز	نوع العملية
1	للتحويل من فهرنهايت إلى مئوية .
2	للتحويل من مئوية إلى فهرنهايت .
3	إنهاء العملية والتوقف .

أدخل رمز العملية المطلوبة والدرجة مفصولتين بفاصلة

ب — وعند اختيار «1» وطباعة درجة الحرارة بالفهرنهايتية 60 :

60 درجة فهرنهايتية تساوى 15.55556 مئوية

ج — وعند اختيار «2» وطباعة درجة الحرارة بالمئوية 32 :

32 درجة مئوية تساوى 89.6 فهرنهايتية .

في المثال شكل (٩ — ٥) تم استخدام الدالة DEF واستدعاؤها للتعامل مع قيمة واحدة ، وب نفس الطريقة يمكننا التعامل مع أكثر من قيمة واحدة ، فعند إيجاد مساحة مستطيل ، فالخطوات والعبارات ستكون كالتالى :

١ — تعريف الدالة لإيجاد المساحة .

٢ — الحصول على قيمتى الطول والعرض .

٣ — استدعاء الدالة لحساب المساحة .

10 DEF FNM (A , B) = A * B

20 LET C = FNM (5 , 8)

30 PRINT C

40 END

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ - بين الأخطاء إن وجدت في كل من العبارات التالية :

10	PRINT	SIN 20	أ
10	LET	A \$ = COS (40)	ب
10	DEF	FNA (B) = B * 2 + 20	ج
10	DEF	(X1 , X2) = (X1 ** 2) - (X1 / X2)	د
10	LET	A = PI + SQR (ABS (20)	هـ

٣ - بين نتائج تنفيذ البرامج التالية :

10	LET	I = _ 2	أ _
20	IF	I > 3 THEN 60	
30	PRINT	INT (I) , INT (I + .5)	
40	LET	I = I + .5	
50	GOTO	20	
60	END		

```

10 DEF      FNA (B) = (B * 1) + 2          ب —
20 READ      B
30 IF      B < 0 TEHEN 70
40 PRINT      FNA (B)
50 DATA      10 , 5 , 2 , - 1 , 4
60 GOTO 20
70 END

```

- ٤ — اكتب برنامجاً يعمل على طباعة جيب الزاوية للزوايا من عشر إلى أربعين .
- ٥ — باستخدام تعليمة DEF اكتب برنامجاً يعمل على الحصول على عشرة أرقام عشوائية ، ومن ثم إيجاد تربيع وتكعيب الرقم .
- ٦ — اكتب برنامجاً يعمل على إيجاد عشرة أرقام عشوائية تقع بين مائة ومائة وخمسين .
- ٧ — اكتب برنامجاً باستخدام الدالة DEF يعمل على :

- (أ) التحويل من بوصة إلى سنتيمتر وبالعكس .
- (ب) التحويل من أميال إلى كيلومترات وبالعكس .
- (ج) التحويل من كجم إلى باوند وبالعكس .

مع العلم بأن :

البوصة = ٢,٥٤ سم ، كجم = ٢,٢٥ باوند ، الميل = ١,٦ كم .

- ٨ — اكتب برنامجاً باستخدام الدالة DEF يعمل على الحصول على ثلاث قيم ، ومن ثم إجراء العمليات الحسابية من جمع وضرب عليها .

الفصل العاشر

التحكم في طبع البيانات المخرجة والمعلومات

استعمالات تعليمة اطبع باستخدام PRINT USING :

تعرضنا في الفصول السابقة لكيفية التحكم إلى حد ما في صيغة النتائج المعالجة ،
والمراد إخراجها باستخدام الفاصلة (,) أو الفاصلة المنقوطة (.) أو تعليمة (TAB).
وكان هذا التحكم في العمود الذي سيبدأ بعده طباعة القيمة التي يحتويها المتغير ، كما
نلاحظ في المثال التالي الذي يقرأ ويطبع خمس قيم :

```
10 REM استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM اظهار كيفية التحكم في الطباعة
50 REM لطباعة العناوين (العمود)
60 PRINT "=====
70 PRINT "      1      2      3      4      5"
80 PRINT "12345678901234567890123456789012345678901234567890"
90 PRINT "=====
100 REM اسناد قيم اوليه للمتغير
110 LET I = 0
120 REM قراءة القيم العددية
130 READ A
140 PRINT TAB(5);A
150 REM اضافة واحد للعداد
160 LET I = I + 1
170 REM استفسار عن قيمة العداد
180 IF I < 5 THEN 130
190 PRINT "=====
200 REM البيانات
210 DATA 45,107,7,320,1320
220 RETURN
```

وفي حالة تنفيذ البرنامج السابق نجد أن المخرجات ستظهر كالتالي :

```

=====
                1                2                3                4                5
12345678901234567890123456789012345678901234567890
=====
      45
     107
      7
    320
   1320
=====

```

نلاحظ أن جميع القيم بدأت طباعتها في العمود السادس كما حدد لها في عبارة رقم 140 ، فلو أردنا إخراج هذه القيم بطباعة كل رقم من كل قيمة في الخانة الخاصة به (خانة الآحاد ، العشرات ، المئات ، الألوف) على حسب قواعد النظام العشري ، أى تحديد الهيئة أو الصيغة (FORMAT) التى ستظهر بها كل قيمة ، يمكننا عمل ذلك باستخدام تعليمة «اطبع باستخدام PRINT USING» ، كما هو موضح في الشكل التالى :

PRINT USING		الشكل العام لعبارة اطبع باستخدام	
X X X	PRINT USING	«X X X X X X»	X ₁ , X ₂ , X ₃ ,
↓	⏟	↓	⏟
رقم السطر	اطبع باستخدام	الصيغة / الهيئة أو متغير يحتوى عليها	اسم المتغير أو أسماء المتغيرات
		F#	

لاحظ أن الصيغة إما أن تظهر في عبارة PRINT USING أو تحزن أولاً في متغير غير حسابى ، ومن ثم يذكر اسم المتغير في العبارة ، وفي كلتا الحالتين فإن الصيغة/الهيئة ستحتوى ما بين علامتى التنصيص « » على رمز خاص أو أكثر ، ويعتمد انتقاء الرموز على نوع البيانات المراد إخراجها (حسابية أو غير حسابية) ، وستتضمن الأمثلة التالية كيفية استخدام هذه التعليمة وفق هاتين الحالتين .

البيانات العددية :

طباعة الأرقام الصحيحة : يستخدم الرمز الخاص « # » لتخصيص كيفية طباعة الأرقام الحسابية . فلإخراج الأرقام السابقة ، كل في خانته الخاصة به ، نعمل على تعديل عبارة 140 في البرنامج السابق فتصبح كالتالى :

```
140 PRINT USING « # # # # # # # # » ; A
```

حيث تم تحديد الحجم الذى ستطبع به كل قيمة بسبع خانات ، كما هو مبين فى تعليمة رقم (140) ، وفى حالة تنفيذ البرنامج ستظهر المخرجات كالتالى :

```
=====
          1          2          3          4          5
12345678901234567890123456789012345678901234567890
=====
      45.000
     107.000
       7.000
     320.000
    1320.000
=====
```

ونجد أن القيمة قد طبعت من اليمين إلى اليسار، ومن ثم وضع كل رقم من كل قيمة فى الخانة الخاصة به .

أما استخدام الأسلوب الآخر فيتطلب ذلك تعديل السطر (140) وإضافة السطر (105) كالتالى :

```
105 LET A$ = « # # # # # # »
```

```
140 PRINT USING A$ ; A
```

طباعة قيم عددية تحتوى على كسور عشرية : فى حالة وجود قيم عددية كسرية يمكننا تعديل الصيغة فى نفس العبارة (140) ، حيث نحدد عدد الخانات الكسرية المراد

طباعتها إلى يمين الفاصلة العشرية («.» باللغة الانجليزية) وعدد الخانات للعدد الصحيح إلى يسارها . ففى المثال السابق ، إذا افترضنا وجود بيانات رقمية تحتوى على كسور عشرية كالتالى :

```
210 DATA 45.0, 107.34.7.9, 320, 1320.714
```

يجب تعديل السطر 140 لتحديد عدد خانات العدد الصحيح وعدد خانات الكسر العشرى ، ليصبح كالتالى :

```
140 PRINT USING « ###.### » ; A
```

وفى حالة تنفيذ البرنامج ، ستظهر المخرجات كالتالى :

```
=====
              1              2              3              4              5
12345678901234567890123456789012345678901234567890
=====
      45
     107
      7
     320
    1320
=====
```

لاحظ أن البيانات فى عبارة رقم (210) احتوت على قيمة صحيحة (320) . لكن احتواء التخصيص على كسور عشرية (« ###.### ») يؤدى إلى إظهار جميع القيم على شكل كسور .

ويمكن تعديل السطر 140 وإضافة السطر 105 مرة أخرى لتتوافق طباعة النتائج مع الأسلوب الآخر .

```
105 LET A$ = « ##### - ### »
140 PRINT USING A$ ; A
```

الرموز الخاصة :

أما طباعة الرموز الخاصة بالقيم المخرجة فيتم كالتالى :
مثال ١ - إظهار إشارة القيمة (سواء كانت سالبة - أو موجبة +) أمامها . ويتم ذلك بوضع إشارة + أمام الرمز ## فى الصيغة كما هو مبين فى العبارة التالية :

```
10 PRINT USING «+ ## #.#» ; 100.3,98.5,-107.3,17.4,-.75
```

وعند تنفيذ العبارة ستظهر المخرجات كالتالى :

```
+ 100.30 + 98.50 - 107.30 + 17.4 - .75
```

لاحظ أنه تم وضع فراغين فى الصيغة حتى يتم تفريق طباعة القيم بعضها عن بعض .

مثال ٢ - طباعة علامة \$ أمام القيم المراد إخراجها ، ويتم بوضع علامتى \$\$ فى بداية الصيغة كالتالى :

```
20 PRINT USING «$$ ## #.#» ; 720.5 , 315.75 , 23.45
```

وعند تنفيذ العبارة ستظهر المخرجات كالتالى :

```
$ 720.50 $ 315.75 $ 23.45
```

مثال ٣ - وضع علامة * أمام القيم المراد إخراجها ، ويتم بوضع علامتى * * فى بداية الصيغة كالتالى :

```
30 PRINT USING «** ## #.#» ; 1720.5 , 23.45
```

وعند تنفيذ العبارة ستظهر المخرجات كالتالى :

```
* 1720.50 *** 23.45
```

لاحظ أنه تمت تعبئة الفراغات السابقة للقيمة بعلامات * في حالة كون حجم القيمة أقل من حجم الصيغة المراد استخدامها ، كما هو مبين في طباعة القيمة 23.45 . أما في حالة كون الرقم المراد طباعته مكوناً من خمسة أرقام ، فلا يتم طباعة «*» . أمام الرقم .

مثال ٤ — لتسهيل قراءة قيم كبيرة ، يمكننا فصل كل ثلاث خانات (أو أقل أو أكثر) بعضها من بعض بفاصلة كالتالي :

40 PRINT USING « ##,## ##,## ##.## » ; 14578.5 , 479621

وعند تنفيذ العبارة ستظهر المخرجات كالتالي :

14 , 578.50 479 , 621.00

وفي جميع الأمثلة السابقة ، يمكن اتباع الأسلوب الثاني كما في المثالين السابقين . ففي المثال الأخير مثلاً نعدل السطر 40 ونضيف السطر 30 كالتالي :

30 LET B\$ = « ##,## ##,## ##.## »

40 PRINT USING B\$: 14578.5 , 479621

البيانات غير العددية :

يتم التحكم في طباعة البيانات غير الحسابية باستخدام نفس التعليمة PRINT USING ولكن باختلاف الرمز المستخدم ، حيث تستخدم (الشرطة المائلة / SLASH) لتحديد البداية والنهاية (حجم المتغير أو عدد الخانات المراد حجزها حتى تتم طباعة قيم المتغير بها) ، وتحسب الشرطة من ضمن هذا العدد ، فلو أردنا طباعة ١٠ أحرف من متغير ، نترك ٨ خانات فارغة بين الشرطتين .

```

10  READ
15  IF N$ = «X X X» THEN 40
20  PRINT USING «/          /» ; N$
30  DATA «X X X» , «سامى على» , «بسام حداد»
40  END

```

وعند تنفيذ البرنامج ، تظهر النتائج كالتالى :

بسام حداد

سامى على

ويمكن تعديل البرنامج السابق كالتالى ؛ ليتوافق مع الأسلوب الثانى لاستخدام

PRINT USING

```

5    LET K$ = «/          /»
20   PRINT USING K$ ; N$

```

وإذا زاد عدد الحروف المخزنة على عدد الحروف المحددة فى عبارة «اطبع باستخدام» يتم تجاهل الزائد المخزن ، أما إذا كان الوضع على العكس من ذلك فيتم ملء الفائق فى عدد الخانات المحدد بالفراغات .

العناوين التعريفية :

يمكن وضع عناوين تعريفية فى تعليمة PRINT USING لإظهارها فى كل مرة يتم فيها استخدام التعليمة ، كما هو مبين فى البرنامج التالى :

```

10 REM استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM اظهار كيفية التحكم في الطباعة باستخدام تعليمة اطبع باستخدام
50 REM اسناد القيم اوليه للمتغير
60 LET I = 0
70 REM اسناد صيغة الطباعة للمتغير I
80 LET F# = " a = ##### b = #####"
90 READ A,B
100 LET I = I + 1
110 PRINT USING F#;A,B
120 DATA 45,9087,4,765
130 REM استفسار عن عدد مرات الطباعة المراده
140 IF I < 2 THEN 90
150 RETURN

```

وفي حالة تنفيذ البرنامج ستظهر المخرجات كالتالى :

a =	45	b =	9087
a =	4	b =	765

لاحظ أن «A #» و «B #» قد عوملتا كعنوانين تعريفين ، ولم تؤثرا على سير البرنامج ، كذلك أخذت بعين الاعتبار، الفراغات الواردة فى الصيغة ، وأخيراً فإن الأسلوب الثانى قد استخدم فى هذا البرنامج .

ملاحظات على استخدام تعليمة PRINT USING :

- ١ — طباعة القيم العددية تتم من اليمين إلى اليسار .
- ٢ — يجب أن يكون عدد أرقام القيم العددية مساوياً أو أقل من عدد الرموز المستخدمة ، وفى حالة زيادة عدد الخانات المراد طباعتها على عدد الخانات المحددة فى التعليمة PRINT USING فتظهر إشارة % على يسار الرقم موضحة هذه الحالة .
- ٣ — يجب أن يكون عدد حروف القيم غير العددية مساوياً أو أقل من عدد المسافات الموجودة بين علامتى \ \ .

- ٤ - يستخدم الرمز # للقيم العددية فقط .
- ٥ - يستخدم الرمز \ \ للقيم غير العددية فقط .
- ٦ - عند استخدام علامة * أو علامة الدولار \$ يتم حجز خانة الإشارة ونحصر من حجم الصيغة ، لذا يجب أن يكون عدد أرقام القيمة أقل من عدد الرموز بواحد .
- ٧ - يفضل استخدام الأسلوب الأول في تحديد الصيغة - أى إدراجها في عبارة PRINT USING إذا كانت العبارة ستنفذ مرة واحدة أو أن الصيغة مبسطة ، أما في حالة استخدام نفس الصيغة في أكثر من عبارة PRINT USING أو أنها معقدة ، فيفضل استخدام الأسلوب الثانى - أى تخزين الصيغة في متغير حرفى .
- ٨ - لاحظ أن استعمال تعليمة «اطبع باستخدام» يكون أفضل ما يكون عند تحديد الصيغة لطباعة كل سطر على حدة ، وسنبين أسلوباً فيما بعد عن كيفية توجيه الطباعة على الشاشة في أى مكان منها .

تطوير برنامج باستخدام PRINT USING :

مثال (١٠ - ١)

الهدف : تعديل مثال (٨ - ١) ليعمل تعليمة اطبع باستخدام PRINT USING بدلاً من TAB .

خطوات الحل : الخطوة الأولى والثانية والثالثة كما في المثال (٨ - ١)

الخطوة الرابعة : البرنامج في الشكل التالى :

شكل (١٠-١)

برنامج لحساب الراتب الأساسي وخارج الدوام باستخدام PRINT USING

```

10 REM      لا ستدعاء برنامج فرعي
20 GOSUB 70
30 END
40 REM      برنامج فرعي لحساب الراتب الا اساسي خارج الدوام و الا جمالي
50 REM      لا ستاد القيم البدائية للمتغيرات و طباعة المتحولات
60 LET T = 0: LET T1=0
70 PRINT TAB(2);"EMPLOYEE NAME";TAB(22);"NO. OF HOURS";TAB(37);"HOUR RATE";TAB(
48);"BASE SALARY";TAB(61);"OVER TIME";TAB(72);"NET PAY"
80 PRINT TAB(2);"=====";TAB(22);"=====";TAB(37);"=====";TAB(
48);"=====";TAB(61);"=====";TAB(72);"=====
90 REM      لقراءة اسم الموظف عدد ساعات العمل واجرة الساعة
100 READ N#,H,P
110 REM      لا ستفسار من نهاية البيانات
120 IF N#="dummy" THEN 290
130 REM      لا ستفسار اذا ممل الموظف خارج دوا
140 IF H <= 40 THEN O = 0: N = H * P
      ELSE O = (H - 40) * (P * 1.5): N = O + (P * 40)
150 REM      لا يحدد الراتب الا اساسي
160 LET B = N - O
170 LET T = T + N
180 LET T1 = T1 + O
190 REM      لطباعة البيانات
200 PRINT USING " \      ##.0      ##.0      ###.0      ##.0
      ",###.##";N#,H,P,B,O,N
210 REM      لقراءة سجل آخر
220 READ N#,H,P
230 GOTO 120
240 DATA ALI SALIH,42,15
250 DATA AHMED AMER,38,19
260 DATA ali ahmed,60,23
270 DATA tameem ahmed,52,18
280 DATA dummy,0,0
290 PRINT TAB(2);"=====
      =====
300 REM      لطباعة اجمالي خارج الدوام و اجمالي رواتب الموظفين
310 LET T# = "TOTAL S      ",###.##
320 PRINT USING T#;T1,T
330 PRINT TAB(2);"=====
      =====
340 RETURN

```

وفي حالة تنفيذ البرنامج السابق، سنحصل على النتائج التالية :

EMPLOYEE NAME	NO. OF HOURS	HOURLY RATE	BASE SALARY	OVER TIME	NET PAY
ALI SALIH	42.0	15.0	600.0	45.0	645.0
AHMED AMER	38.0	19.0	684.0	0.0	684.0
ali ahmed	60.0	23.0	920.0	390.0	1,610.0
tameem ahmed	52.0	18.0	720.0	324.0	1,044.0
			TOTAL S	1,059.0	3,883.0

ملخص عن الرموز المستخدمة في تعليمة PRINT USING :

الرمز والهيغة	شرح لاستخدامه
#####	لطباعة قيم عددية صحيحة قيمتها لا تزيد على 99999 .
#####.	لطباعة قيم عددية غير صحيحة (كسور) قيمتها لا تزيد على 999.99 .
#####.###	لطباعة قيم عددية غير صحيحة ، مع وضع فاصلة بين الأرقام الثلاثة التي توجد على يسار الفاصلة العشرية (.) ، وما بعدها من أرقام وقيمها لا تزيد على 999.99,999 .
\$\$\$#####	لطباعة قيم عددية غير صحيحة مسبقة بعلامة الدولار \$ مع وجود فاصلة بين الأرقام الثلاثة ، التي توجد على يسار الفاصلة العشرية (.) وما بعدها من أرقام وقيمها لا تزيد على 99,999.99 \$.
##### -	لطباعة قيم عددية صحيحة متبوعة بإشارة السالب على يمين القيمة وقيمها لا تقل عن 9999 .
##### *	لطباعة قيم عددية صحيحة مسبقة بإشارة * وقيمها لا تزيد على 9999 .
##### +	لطباعة قيم عددية صحيحة مسبقة بإشارة القيمة ، سواء كانت موجبة أو سالبة ، وقيمها لا تزيد على 99999 .
!	لطباعة الحرف الأول من القيمة غير العددية .
..... 1 2 8	لطباعة قيمة غير عددية وعدد حروفها لا يزيد على ثمانية أحرف . وللاستخدام الأمثل لتعليمات تحديد شكل وصيغ المعلومات المخرجة ، من المفيد أن نتحدث عن تصميم المخرجات والمدخلات .

تصميم المخرجات و المدخلات :

المخرجات :

في معرض الحديث عن تعليمات التحكم في طباعة البيانات مثل TAB,PRINT USING,PRINT تم توضيح كيفية طباعة العناوين المعرفة للبيانات ، دون تحديد مسبق لأماكن طباعتها سواء كانت الطباعة على الشاشة أم على الورق ، وعند تصميم برنامج لتطبيق ذى مستوى محدد من التعقيد سيكون من المحتمل تصميم المخرجات من البرنامج بشكل مسبق . و يوضح هذا التصميم العمود والسطر في الصفحة ، أو الشاشة الواحدة والمحدد لبداية ونهاية طباعة كل حقل من البيانات أو من العناوين . وهناك نماذج خاصة لتصميم المخرجات على الورق وأخرى للشاشات . ويحتوى النموذج الأول على ٦٠ سطراً و ١٣٢ عموداً ، ومن الممكن طبعا استخدام عدد أقل من الأسطر أو الأعمدة أو من كليهما . في حين يحتوى نموذج تصميم الشاشات على ٨٠ عموداً و ٢٤ سطراً .

وعند استخدام هذه النماذج في طباعة المعلومات باللغة العربية ، قد تواجه مشكلة بسيطة ، وسبب تلك المشكلة هو التفاوت بين الحروف في عدد الخانات اللازمة لطباعتها ، وفي كل الأحوال لا بأس من الاسترشاد بهذه النماذج ، حيث تزداد الاستفادة منها مع ازدياد الخبرة في طباعة الحروف العربية . هذا مع العلم بوجود جداول توضح عدد الخانات التى يتطلبها كل حرف .

ومن المفيد في معظم الأحيان تحديد مواصفات البيانات التى ستظهر كمخرجات ، والمثال التالى يبين المواصفات التى ستظهر في كشف علامات الدارس ، وسيتم استخدام الحرف x للدلالة على البيانات الحرفية والرقم ٩ للدلالة على الأرقام .

مواصفات البيانات في كشف علامات الدارس

البيانات	الطول	النوع	اسم المتغير
الاسم :	٣٥	حرفي	N X (35)
الفصل :	١٠	حرفي	S X (10)
التاريخ :	٨	رقمي	9999-99-99
	٢	اليوم	D 99
	٢	الشهر	M 99
	٤	السنة	Y 9999
الدورة :	٢	رقمي	D1 99
المواد وعلاماتها متكررة ٥ مرات على الشكل التالي :			
المواد :	٢٥	حرفي	S X (26)
العلامات :	١	حرفي	G X
المعدل :	٤	رقمي	A 9.99

(لاحظ حساب خانة واحدة للفاصلة العشرية)

ويحتوى الشكل التالى تصميماً لكشف علامات الدارس :

رقم العمود

0 10 20 30 40 50 60 70 80

رقم السطر

1	الاسم : x	(35)	x الفصل : x	(10)	y الدورة ٩٩	التاريخ ٩٩ - ٩٩ - ٩٩٩٩
2						
3	المادة			العلامة		
4	x	x	x	x		
5	x	x	x	x		
6	x	x	x	x		
7	x	x	x	x		
8	x	x	x	x		
9	x	x	x	x		
10						
11	المعدل العام				9.99	

لاحظ عدم ضرورة تكرار الحرف x ، حيث يذكر العدد فوق الخط الموصل بين بداية ونهاية الحقل .

أما العبارات الخاصة بطباعة هذا النموذج باستخدام تعليمة PRINT USING فهي كالتالى :

I. P. A

VDU SCREEN LAYOUT

PROGRAM : _____ DATE _____

MAPSET : _____

MAP NAME : _____

COMMENTS : _____
لصيف كذا، درجات على نموذج (صيف)

	10	20	30	40	50	60	70	80
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

الصفحة 1 من 1

الصفحة 2 من 2

الصفحة 3 من 3

الصفحة 4 من 4

الصفحة 5 من 5

الصفحة 6 من 6

الصفحة 7 من 7

الصفحة 8 من 8

الصفحة 9 من 9

الصفحة 10 من 10

الصفحة 11 من 11

الصفحة 12 من 12

الصفحة 13 من 13

الصفحة 14 من 14

الصفحة 15 من 15

الصفحة 16 من 16

الصفحة 17 من 17

الصفحة 18 من 18

الصفحة 19 من 19

الصفحة 20 من 20

الصفحة 21 من 21

الصفحة 22 من 22

الصفحة 23 من 23

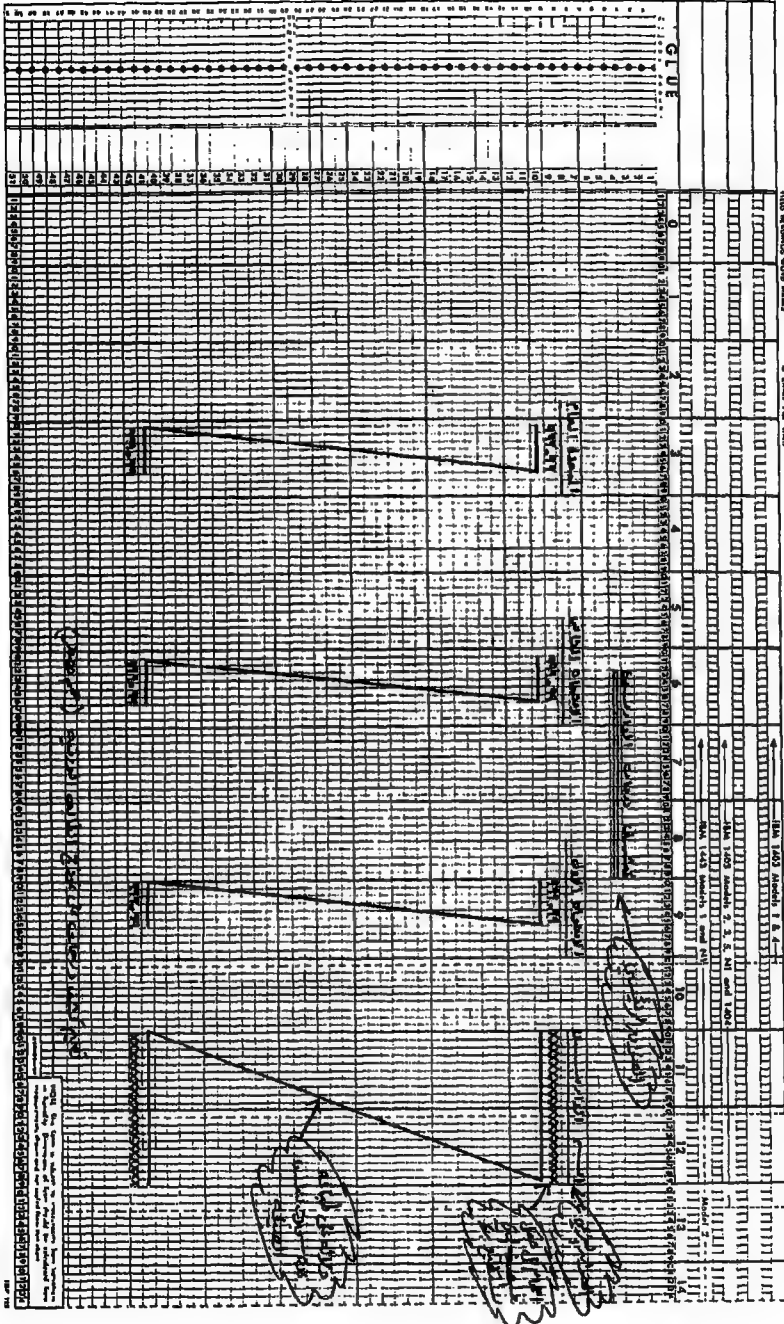
الصفحة 24 من 24

I.P.A

INTER SPACING CHART
1403, 1404, and 1443

Price range

بركي لاسي لاي



المدخلات :

تم استخدام تعليمة أدخل INPUT لتسلم البيانات من الشاشة . ولاحظنا أنه عند تنفيذ هذه التعليمة ، يتم إظهار علامة الاستفهام ؟ يعلمنا أن البرنامج يكون في حالة انتظار للمدخلات ، ويتم إيضاح كيفية طباعة التعليمات الخاصة بالبيانات المدخلة حتى يعلم مستخدم النظام طبيعة هذه البيانات ، ويتم فعل ذلك كله في المكان الذي يلي تنفيذ آخر تعليمة ، فلو توألى تنفيذ تعليمات INPUT لكان ظهور علامات الاستفهام «؟» متتالية على الشاشة حسب تسلسل خطوات البرنامج ، وقد تتخللها طباعة نتائج جزئية مرحلية .

ويمكن استخدام تعليمة CLS قبل تنفيذ تعليمة INPUT بحيث تظهر علامة الاستفهام — سيقوم بأية رسائل إيضاحية أو أوامر للإدخال — على رأس الشاشة في الزاوية اليمنى ... وهكذا ، وكذلك استخدام نفس التعليمة (CLS) قبل تعليمة PRINT يؤدي إلى طباعة المخرجات بدءاً من أول سطر للشاشة ... وهكذا .

وفي هذا الجزء يهمننا نوعان رئيسيان من المؤشرات :

الأول — المؤشرات : تأخذ المؤشرات في العادة قيمة واحدة 1,2,3,4 ... أو حرفاً واحداً A,b,c,...X . وهى تحدد الجزء من البرنامج الذى سيتشعب إليه لتنفيذ معالجة معينة . وهذه عادة ما تظهر في الجزء المتحكم في عدة أجزاء تنفيذية . ويتم في العادة تصميم هذه الشاشات — شاشات الاختبارات — بشكل هرمى يتمشى مع الهيكل الهرمى الوظيفى للبرنامج ، كما تم شرحه في الفصل الرابع .

الثانى — البيانات : حيث يتم تصميم شاشات خاصة باستقبال البيانات . وهذه البيانات المدخلة قد تكون خاصة بالموظف أو بقطع المخزون ، أو بالدارسين ... إلخ .

أما المخرجات من البرنامج فيصمم لها شاشات كما مر ذكره ، وكذلك لطباعة
العناوين والبيانات .

التحكم فى استقبال أو طبع المعلومات فى أى مكان على الشاشة :
فالتحكم المطلوب إذن هو إمكانية استقبال أو طبع البيانات فى أى مكان فى شاشة
العرض ، أو عند الضرورة ، الطبع فقط على الطابعة وتقسّم الشاشة إلى ٢٥ سطراً* و
٨٠ عموداً ، أى أن هناك ٢٠٠٠ موقع يمكن بدء إدخال أو طباعة المعلومات منها .
وحتى نفهم كيفية توجيه المؤشر CURSOR إلى أى من هذه المواقع (الخانات) ، يمكن
تخيل قلم يتم نقله من موقع لآخر دون أن يؤثر ذلك على ما كتب على الشاشة ، إلا أن
تطمس الكتابة السابقة بكتابة جديدة .

والتعليمة التى تستخدم لنقل المؤشر (القلم الوهمى) إلى الخانة المطلوب إدخال
البيانات أو طباعتها منها ، هى تعليمة حدد (موقع) LOCATE .

وتتبع الشكل العام التالى :

الشكل العام لعبارة LOCATE	
<p>XXX</p> <p>↓</p> <p>رقم السطر</p>	<p>LOCATE</p> <p>↓</p> <p>التعليمة</p>
<p>XX</p> <p>↓</p> <p>١) رقم السطر (ROW)</p>	<p>XX</p> <p>↓</p> <p>٢ - رقم العمود (COLUMN)</p>
<p>الخلية لإظهار البيانات فيه أو استقبالها منه</p>	

• لاستخدام السطر الخامس والعشرين لابد من إزالة التعليمات الخاصة باستعمال لغة بيسك منها وذلك باستخدام
التعليمة التالية قبل محاولة طباعة أى معلومات : XXX KEY OFF

١) يجب ألا يزيد رقم السطر في حالة استخدام الشاشة على ٢٥ أو ٧٢ في حالة استخدام الطابعة .

٢) يجب ألا يزيد رقم العمود في حالة استخدام الشاشة على ٨٠ أو ١٣٢ في حالة استخدام الطابعة .

ولتوضيح كيفية عمل تعليمة حدد LOCATE ، سيتم طباعة قيمة متغير في كل من الزوايا الأربع للشاشة ومنتصفها .

فالخطوات المطلوبة هي :

- ١ — إسناد قيمة المتغير .
- ٢ — تحديد الزاوية اليسرى العليا وطباعة القيمة .
- ٣ — تحديد الزاوية اليمنى العليا وطباعة القيمة .
- ٤ — تحديد الزاوية اليسرى السفلى وطباعة القيمة .
- ٥ — تحديد الزاوية اليمنى السفلى وطباعة القيمة .
- ٦ — تحديد منتصف الشاشة وطباعة القيمة .

العبارة المستخدمة لكل من الخطوات المذكورة سابقاً :

10 LET A = 20	الخطوة رقم ١
20 LOCATE 1,1: PRINT	الخطوة رقم ٢
30 LOCATE 1,70: PRINT A	الخطوة رقم ٣
40 LOCATE 24,1.: PRINT A	الخطوة رقم ٤
50 LOCATE 24,70: PRINT A	الخطوة رقم ٥
60 LOCATE 12,35: PRINT A	الخطوة رقم ٦

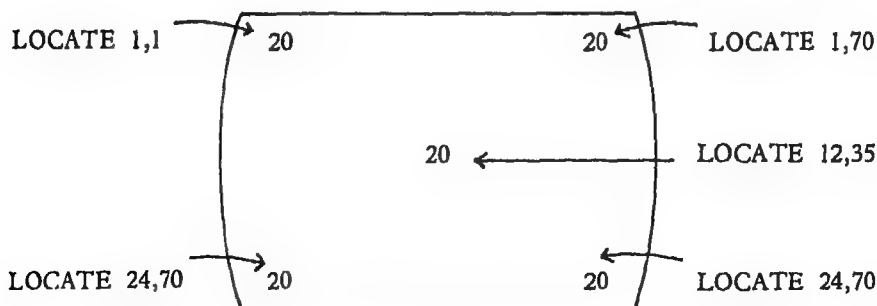
وفي حالة تنفيذ هذه التعليمات سيتم إظهار قيمة المتغير A (20) في الزوايا الأربع ،
ومنتصف الشاشة مع أى محتويات أخرى كانت ظاهرة على الشاشة سابقاً ، وللحيلولة
دون الاحتفاظ بأية بيانات أخرى على الشاشة مع البيانات المراد إظهارها نستخدم
تعليمية CLS (لمسح الشاشة CLEAR THE SCREEN) كما هو مبين في عبارة رقم 60
في البرنامج التالى والموضح للخطوات السابقة .

```

10 REM      لا استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM      برنامج فرعي لطباعة قيمه في زوايا الشاشة
50 REM      و من ثم في منتصفها
60 REM      مسح محتويات الشاشة و اسناد قيمه للمتغير
70 CLS: LET A = 20
80 REM      تحديد الزاويه العلويه اليسرى وطباعة القيمه
90 LOCATE 1,1 : PRINT A
100 REM      تحديد الزاويه العلويه اليمنى وطباعة القيمه
110 LOCATE 1,70: PRINT A
120 REM      تحديد الزاويه السفلى اليسرى وطباعة القيمه
130 LOCATE 24,1 : PRINT A
140 REM      تحديد الزاويه السفلى اليمنى وطباعة القيمه
150 LOCATE 24,70: PRINT A
160 REM      تحديد منتصف الشاشة و طباعة القيمه
170 LOCATE 12,35: PRINT A
180 RETURN

```

وفي حالة تنفيذ البرنامج نجد أن البيانات ستظهر كالتالى :



ملاحظة :

يجب أن يؤخذ بعين الاعتبار حجم البيانات المراد إظهارها ، وعلى ضوء ذلك يحدد المكان/المركز ، الذى ستبدأ منه طباعة هذه البيانات ، و يتحقق ذلك عن طريق تصحيح الشاشات كما مر ذكره سابقاً في هذا الفصل . وكما هو مبين في العبارتين (100 ، 140) كان حجم قيمة المتغير A معروفاً لدينا و يساوى (٢) وعدد خانات الشاشة (٧٢) ؛ لذا نبدأ في الطباعة من خانة/عمود رقم (٧٠) ، وإلا سيتوقف البرنامج بسبب اجتياز القيمة النهائية لعدد الخانات الممكن استخدامها .

وينطبق هذا الشرط أيضاً على عدد الأسطر الممكن استخدامها ، كما هو مبين في العبارتين (120 و 140) .

لائحة الاختيارات الهرمية وتصميم الشاشات :

تتسم معظم البرامج المطورة حالياً بخاصية التفاعل والتخاطب فيما بين البرنامج والمستفيد المستخدم ، وذلك عن طريق شاشة العرض ولوحة المفاتيح .
وحيث إن تسلسل تنفيذ الأجزاء الخاصة بأى برنامج يخضع للهيكل الهرمى ،
الوظيفى ، فإنه من المحتم تطوير البرنامج التخاطبى عن طريق تصميم شاشات ترتبط فيما بينها بشكل هرمى متناسب ومتوافق مع الهيكل الهرمى للأجزاء .

أنواع ووظائف الشاشات الرئيسية :

أولاً - شاشات لائحة الاختيارات : وظيفة هذه الشاشات أن تحدد الاختيارات المتاحة للمستخدم ، في أى مرحلة من مراحل تنفيذ أجزاء البرنامج والرموز التى ستمثل كل اختيار . وستكون هذه الشاشات مرتبة هرمياً حسب تسلسل تنفيذ الأجزاء الظاهرة في الهيكل الهرمى للبرنامج .

و يتم التشعب وفق هذه الخيارات إلى الجزء ذى العلاقة ، والذي من الممكن أن يقوم بواحدة أو أكثر من الوظائف التالية : عرض لائحة اختيارات أخرى في مستوى أقل في الهيكل الهرمى للشاشات ، طلب إدخال بيانات ، معالجة حسابية أو منطقية أو كليهما ، طباعة تقرير إما على الشاشة أو على الورق .

ثانياً — شاشات الإدخال : تتطلب هذه الشاشات إدخال بيانات ذات نوعين :

— بيانات تفصيلية عن سجل شيء معين ، مثل : قطع المخزون ، الموظف ، الدارس ، كالا اسم والرقم والعنوان والدورة ... إلخ .

— بيانات للقيام بالاستفسار عن موظف ، مثل طباعة اسم أو رقم الدارس . ومن ثم عرض المعلومات المخزنة عنه بعد البحث عن سجل الدارس المحدد في ملف الدارسين .

ثالثاً — شاشات الإخراج : تقوم هذه الشاشات بطباعة المعلومات الناتجة عن البرنامج ، أو المعالجة ، أو الاستفسارات ذات العلاقة بالبرنامج ، كعرض المعلومات عن قطعة معينة أو إحصائيات ملخصة .

وسنركز في هذا الفصل على استعمال تعليمة حدد (موقع) LOCATE لعرض شاشات الاختيارات إذ أن استخدام هذه التعليمة في شاشات الإدخال والإخراج غير فعال .

تطوير برنامج باستخدام لائحة الاختيارات الهرمية :

مثال (١٠ - ٢) :

الهدف : تطوير برنامج (مثال ٨ - ٥) باستخدام لائحة الاختيارات ، على أن يقوم بالاستفسارات التالية :

— الحصول على الدخل الكلى للشركة .

- الحصول على دخل فرع معين .
- النسبة المئوية لدخل كل من الفروع إلى الدخل الكلى للشركة .

أولاً — خطوات الحل :

- ١ — إظهار لائحة الاختيارات الرئيسية ، والتي بواسطتها يتم اختيار نوع الاستفسار المطلوب .
- ٢ — إذا كان الاختيار يساوى (١) نحسب دخل كل سيارة من السيارات المؤجرة ، ونضيفه إلى المجموع الكلى للشركة ، ونستمر إلى نهاية البيانات ، ومن ثم نظهر المجموع الكلى للشركة .
- ٣ — إذا كان الاختيار يساوى (٢) نبحث عن السيارات المؤجرة فى ذلك الفرع ، ونحسب دخل السيارة ، ونضيفه إلى المجموع الكلى لذلك الفرع ، ونستمر إلى نهاية البيانات ، ومن ثم نظهر المجموع الكلى للفرع .
- ٤ — إذا كان الاختيار يساوى (٣) نحسب دخل كل سيارة ومن ثم نضيف الناتج إلى كل من الدخل الكلى للفرع الذى استؤجرت منه والدخل الكلى للشركة ، ونستمر إلى نهاية البيانات ، ومن ثم نحسب النسبة المئوية بضرب الدخل الكلى لكل فرع فى مائة ، ثم قسمة الناتج على المجموع الكلى لدخل الشركة .

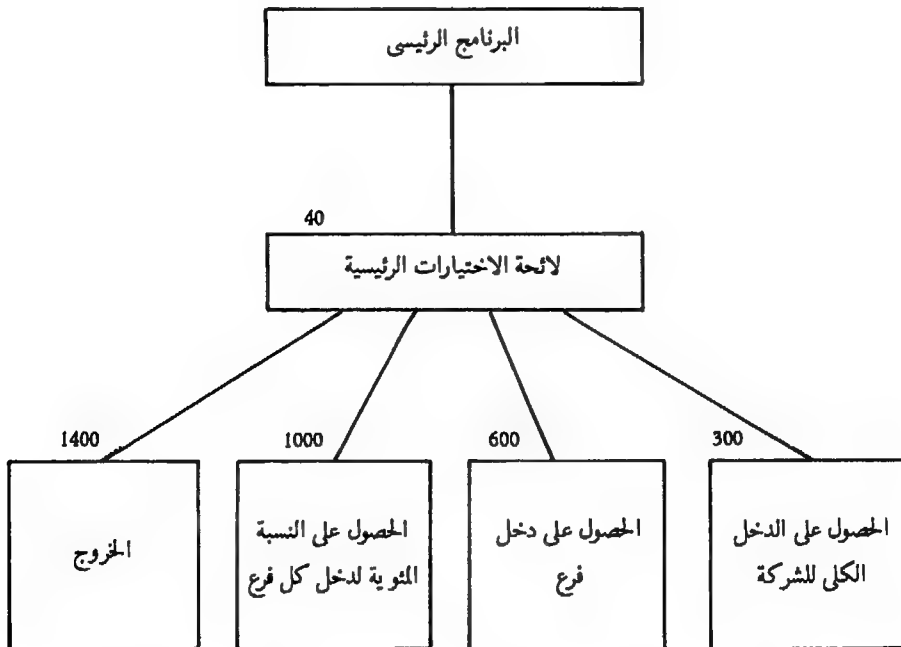
ملاحظة :

لإجابة كل من الاستفسارات الثلاثة يجب أن نتعامل مع جميع البيانات فى كل مرة يتم فيها الاستفسار عن أى من الحالات .
لكن بعد كل قراءة قيمة من البيانات يتم انتقال المؤشر إلى بداية القيمة التى تليها ، إلى أن يصل إلى نهاية البيانات المزودة فى البرنامج . وإذا أردنا استخدام هذه البيانات فى عملية أخرى يجب أن نعيد المؤشر إلى بداية البيانات المزودة ويمكننا عمل ذلك باستخدام تعليمة (أعد التخزين RESTORE) .

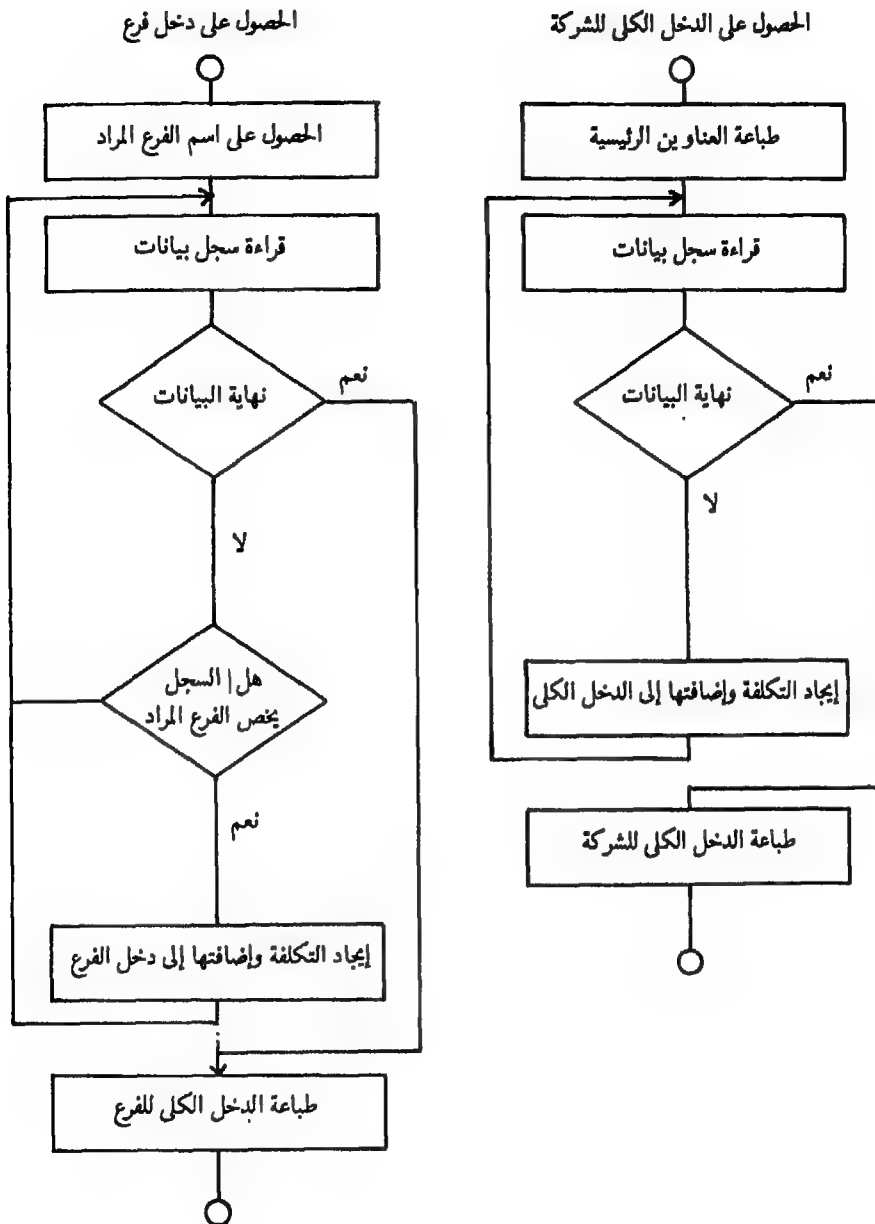
الشكل العام لعبارة	RESTORE
التعليمة	RESTORE
رقم السطر	XXX

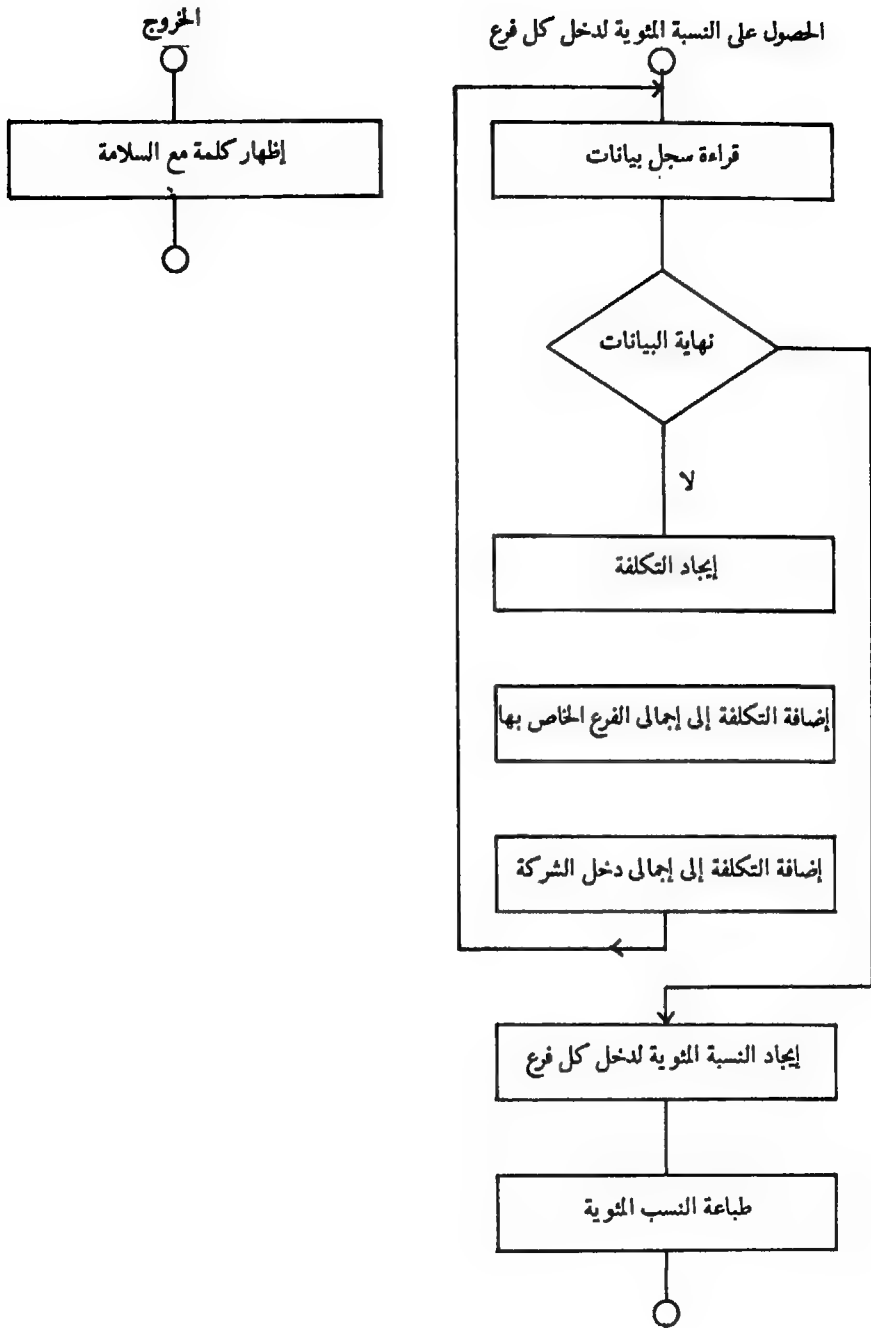
ويتم استخدامها في البرنامج عندما ننهي قراءة البيانات المراد معالجتها ، ونريد التعامل مع هذه البيانات من بدايتها .

ثانياً - الهيكل الهرمي :



ثالثاً - مخطط البرمجة التركيبية :





رابعاً - البرنامج في الشكل التالي :

شكل (١٠-٢)

برنامج لإيجاد الدخل الكلي لكل فرع من فروع الشركة والدخل الإجمالي لشركة

```

10 REM البرنامج الرئيسي
20 REM استعداء برنامج فرعي للاظهار لا فحة الاختيارات ومن ثم التشغيل
30 GOSUB 50
40 END
50 REM برنامج فرعي للاظهار لا فحة الاختيارات
60 CLS
70 LOCATE 6,30: PRINT "الا فحة الاختيارات الرئيسية"
80 LOCATE 7,30: PRINT "-----"
90 LOCATE 8,30: PRINT "نوع العملية"
100 LOCATE 9,30: PRINT "الرمز"
110 LOCATE 10,30: PRINT "للحصول على دخل الشركة الا جمالي"
120 LOCATE 11,30: PRINT "للحصول على دخل احد الفروع"
130 LOCATE 12,30: PRINT "للحصول على النسبة المئوية لكل فرع"
140 LOCATE 13,30: PRINT "لا فحة العمليات"
150 LOCATE 14,30: PRINT "-----"
160 LOCATE 16,30: PRINT "ادخل رمز العملية المراد"
170 LOCATE 16,30: INPUT N
180 REM لا ستفسار عن نوع العملية
190 ON N GOSUB 260,660,1050,1460
200 REM لا ستفسار عن نهاية العمليات وصحة البيانات المدخلة
210 IF N = 4 THEN 250
220 IF N > 0 AND N < 4 THEN 60
230 LOCATE 20,25: PRINT "***** خط في الا دخال *****"

```

```

240 GOTO 170
250 RETURN
260 REM -----
270 REM برنامج فرعي للاظهار لا ثمة الاختيارات
280 REM -----
290 REM لطباعة المعناوين
300 CLS:PRINT TAB(30);"الدخل الكلي للمشركه"
310 LET T1$="زيادة تكلفه"
    الـ
320 LET T2$="
330 LET T3$="=====
=====
340 PRINT " اسم الزبون عند عداد عند كم ايام زياده تكلفه
    الـ
350 PRINT "
360 PRINT "=====
=====
370 ' PRINT USING T1$: PRINT USING T2$: PRINT USING T3$
380 LET T = 0: REM اسناد القيمه الا وليه للدخل الكلي
390 RESTORE: REM وضع المؤشر على بدايه البيانات
400 REM لقراءه البيانات
410 READ C,P,N$,M1,M2,D
420 REM للاستفسار عن مؤشر نهايه البيانات
430 IF C = 4 THEN 570
440 GOSUB 1330: REM الحساب للتكلفه
450 REM لمسابي التكلفه الكليه
460 LET A = A1 + A2
470 REM اضافه التكلفه للدخل الكلي للمشركه

```



```

710 LOCATE 9,30:PRINT "=====
720 LOCATE 10,30:PRINT "      لا تُعده أسماء الطر وع المتوفره
730 LOCATE 11,30:PRINT "=====
740 LOCATE 12,30:PRINT "      اسم الطر وع
750 LOCATE 13,30:PRINT "      -----
760 LOCATE 14,30:PRINT "      الرياض
770 LOCATE 15,30:PRINT "      جده
780 LOCATE 16,30:PRINT "      الدمام
790 LOCATE 17,30:PRINT "=====
800 LOCATE 19,30:PRINT "      ادخل رمز الطر وع
810 LOCATE 19,30:INPUT B
820 REM
830 IF B > 0 AND B < 4 THEN 860
840 LOCATE 20,30:PRINT "*****
850 GOTD 810
860 REM
870 RESTORE:READ C,P,N$,M1,M2,D
880 LET F1 = 0
890 REM
900 IF C = 4 THEN 970
910 IF C < > B THEN 950: REM المطلوب الطر وع
920 GOSUB 1330: REM
930 LET F1 = F1 + A1 + A2
940 REM
950 READ C,P,N$,M1,M2,D
960 GOTD 890
970 REM
980 IF B = 1 THEN B$= "الرياض"
      طباعه المدخل الكلي للطر وع

```

```

990 IF B = 2 THEN B$ = "جده"
1000 IF B = 3 THEN B$ = "الدمام"
1010 CLS:LOCATE 12,30
1020 PRINT F1;" = ";B$;"
1030 FOR W=1 TO 9000:NEXT W
1040 RETURN
1050 REM -----
1060 REM برنامج فرعي لحساب النسبة المئوية لدخل كل من فروع الشركة
1070 REM -----
1080 REM المثال ٣
1090 LET F1=0: LET F2 = 0: LET F3 = 0: LET T = 0
1100 REM وضع المؤشر في بداية البيانات وقراءة السجل
1110 RESTORE: READ C,P,N$,M1,M2,D
1120 REM للا ستفسار عن مؤشر نهايه البيانات
1130 IF N <> 3 AND C = 4 THEN 1320
1140 IF C = 4 THEN 1230
1150 REM استدعاء برنامج فرعي لحساب التكليفه
1160 GOSUB 1330: LET T = T + A1 + A2
1170 IF C = 1 THEN F1 = F1 + A1 + A2
1180 IF C = 2 THEN F2 = F2 + A1 + A2
1190 IF C = 3 THEN F3 = F3 + A1 + A2
1200 REM قراءة سجل اخر
1210 READ C,P,N$,M1,M2,D
1220 GOTO 1140
1230 REM لحساب النسبة المئوية لكل من الفروع المثال ٣ وطباعتها
1240 LET T5$="#.#.#" = \
1250 LET P1 = ( F1 * 100 ) / T
1260 LET P2 = ( F2 * 100 ) / .T

```



```

1270 LET P3 = ( F3 * 100 ) / T
1280 CLS:LOCATE 10,20:PRINT USING T5$;P1,"
الرياضي ",P2,"
جده "
1290 LOCATE 11,20:PRINT USING T5$;P2,"
جده "
1300 LOCATE 12,20:PRINT USING T5$;P3,"
الدمام "
1310 FOR W = 1 TO 3000: NEXT W
1320 RETURN
1330 REM -----
1340 REM
1350 REM
1360 REM
1370 LET M = M2 - M1
1380 REM
1390 LET A1 = D * 100
1400 REM
1410 LET K = D * 150
1420 REM
1430 LET A2 = 0
1440 IF M > K THEN A2 = ( M - K ) * 3.5
1450 RETURN
1460 REM -----
1470 REM
1480 REM
1490 REM
1500 CLS
1510 LOCATE 10,30:PRINT "=====
"
1520 LOCATE 11,30:PRINT "
مع السلا م
"
1530 LOCATE 12,30:PRINT "=====
"
1540 RETURN

```

وفي حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

أ - إظهار لائحة الاختيارات الرئيسية وإدخال «1» للاختيار الأول :

لائحة الاختيارات الرئيسية

الرمز	نوع العملية
1	للحصول على دخل الشركة الإجمالي
2	للحصول على دخل أحد الفروع
3	للحصول على النسبة المئوية لكل فرع
4	لإنهاء العمليات
أدخل رمز العملية المرادة	
?	1

فتظهر الشاشة التالية :

المدينة	اللوحة	اسم الذبون	الدخل الكلي للحركة	كم	أيام	زيادة	تكلفة
الرياض	13482	ali	1680	2220	3	8195.0	8495.0
جدة	42901	tameem	8365	1195	5	1557.5	2057.5
الدمام	81479	ahmed	4712	382	6	0.0	800.0
مجموع الدخل الكلي للحركة = 9152.5							

ب - وإذا دخل رقم «2» من لائحة الاختيارات الرئيسية، تظهر الشاشة التالية :

لائحة أسماء الفروع المتوفرة

الرمز	اسم الفرع
1	الرياض
2	جدة
3	الدمام
أدخل رمز الفرع	
?	1

فتظهر الشاشة التالية :

الدخل الكلى لفرع الرياض = 6495

جـ- إذا أدخل الرقم «3» من لائحة الاختيارات الرئيسية، تظهر الشاشة التالية :

النسبة المئوية لدخل فرع	الرياض	=	70.96
النسبة المئوية لدخل فرع	جدة	=	22.48
النسبة المئوية لدخل فرع	الدمام	=	6.56

وإذا أدخل الاختيار رقم «4» ستظهر الشاشة التالية :

: مع السلامة :

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ - بين الأخطاء إن وجدت في كل من العبارات التالية :

10	PRINT USING	" / / " , A	أ
10	PRINT USING	" # # , # # # " , Y	ب
10	PRINT USING	# # # , B	ج
10	PRINT USING	" # . # # # . # # # " , A,B	د
10	PRINT USING	" - \$ \$, # # # . # # # " , R	هـ

٣ - لأى من البيانات تستخدم الرموز التالية في تعليمة PRINT USING

#	أ
/	ب
*	ج
!	د
\$	هـ

٤ - اكتب برنامجاً يعمل على استقبال رقمين من الشاشة ، ومن ثم إظهار لائحة الاختيارات الهرمية لإعطاء المستخدم إمكانية إجراء أى من العمليات الرياضية التالية عليهما :

- أ - الجمع .
- ب - الضرب

جـ - طرح القيمة الأولى من الثانية.

د - طرح القيمة الثانية من الأولى

ومن ثم إظهار ناتج العملية . والاستفسار عن الاستمرارية في استخدام البرنامج .

ملاحظة : استخدم تعليمة LOCATE

هـ - اكتب برنامجاً يعمل على قراءة خمسة سجلات ، مع العلم بأن حقول كل سجل كالتالى : اسم البائع (عشرة أحرف) ، عدد القطع المباعة من الصنف الأول ، عدد القطع المباعة من الصنف الثانى ، ومن ثم إيجاد التالى :

(أ) الدخل مع العلم بأن :

— سعر القطعة من الصنف الأول ٢٠٠ ريال .

— سعر القطعة من الصنف الثانى ٢٥٠ ريالاً .

(ب) نصيب البائع من الأرباح ، مع العلم بأن النسبة تعتمد على الدخل وموزعة كالتالى :

— ١٠% إذا كان الدخل أكثر من ١٠٠٠٠ ريال .

— ٨% إذا كان الدخل يتراوح ما بين ٨٠٠٠ — ١٠٠٠٠ ريال .

— ٦% إذا كان الدخل يتراوح ما بين ٥٠٠٠ — ٧٩٩٩ ريالاً .

— ٤% إذا كان الدخل يتراوح ما بين ٣٠٠٠ — ٤٩٩٩ ريالاً .

— صفر إذا كان الدخل أقل من ٣٠٠٠ ريال .

(جـ) طباعة النتائج كالتالى :

اسم البائع ، عدد القطع المباعة من الصنف الأول ، الدخل من الصنف الأول ، نصيب البائع من الأرباح ، عدد القطع المباعة من الصنف الثانى ، الدخل من الصنف الثانى ، نصيب البائع من الأرباح ، إجمالى نصيب البائع من الأرباح .

ملاحظة :

اعمل على اختيار عناوين رئيسية وفرعية مناسبة للتقرير ونتائج المعالجة (أى تصميم نموذج مناسب للطباعة) .

٦- يمكن العودة إلى أى من البرامج فى الفصول السابقة ، وإعادة طباعة المخرجات باستخدام تعليمة PRINT USING .

٧- يحتوى سجل البرامج فى معهد الإدارة العامة على المعلومات التالية :

— رقم البرنامج أربعة أرقام

— رمز التخصص حاسب آلى 01 بنوك 06

مالية 02 مكاتبات 07

ميزانية 03 سكرتارية 08

مستودعات 04

أنظمة 05

— مدة التدريب فصل واحد (١) أو فصلان (٢) أو أربعة فصول (٤)

— مستوى الدارسين كفاءة 1

ثانوية 2

جامعى 3

— الدرجة المعين عليها الدارس ٣ أو ٤ أو ٦

— الدرجة المعين عليها بعد التدريب ٤ أو ٨

— تاريخ بدء الدورة رقم الفصل ١ ، ٢

السنة x ١٤٠

اليوم xx

الشهر xxx

- مجموع الساعات للدورة ٦٠ أو ١٢٠ أو ٤٥
— عدد الدارسين في الدورة ١٠ — ٥٠
اعمل على تطوير برامج وفق الهيكل الهرمي للاختيارات للحصول على الإحصائيات التالية :
- أ — مجموع الدارسين حسب البرنامج (استعمل مصفوفة ذات بعد واحد) .
ب — عدد البرامج حسب مستوى الدارسين .
ج — عدد البرامج حسب فصل البداية (الأول أو الثاني) .
استخدم البيانات التالية في البرنامج ، ويمكن إضافة بيانات أخرى .

1123 , 01 , 1 , 1 , 3 , 4 , 1 , 1407 , 10 , 2 , 60 , 15
1231 , 05 , 3 , 3 , 6 , 8 , 1 , 1407 , 10 , 2 , 60 , 20
1251 . 03 , 2 , 3 , 6 , 7 , 1 , 1407 , 10 , 2 , 45 , 25
1121 , 01 , 4 , 2 , 4 , 6 , 2 , 1408 , 5 , 6 , 120 , 25
2311 , 02 , 2 , 3 , 6 , 7 , 2 , 1408 , 5 , 6 , 45 , 20
3112 , 07 , 2 , 3 , 6 , 7 , 2 , 1408 , 5 , 6 , 45 , 25
1121 , 08 , 2 , 2 , 4 , 6 , 1 , 1407 , 10 , 2 , 60 , 20

المصفوفات ذات البعد الواحد وتعليمات الدوارة البسيطة

مقدمة عن المصفوفات ذات البعد الواحد :

تحدثنا في الفصول السابقة عن استخدام أسماء المتغيرات لتشير إلى صناديق مستقلة داخل ذاكرة الحاسب ، يتم تخزين البيانات فيها ، ومن الأمثلة على ذلك :

N\$
<input type="text"/>
A
<input type="text"/>
S
<input type="text"/>

تخزين أسماء الدراسين

تخزين العمر

تخزين مساحة المستطيل

وفي الأمثلة السابقة لاحظنا أن القيمة الحالية لمحتويات الصندوق من البيانات تمحى من الذاكرة عند تخزين القيمة الجديدة لها .
ففى المثال البسيط التالى يتم قراءة الاسم وطباعته إلى حين الانتهاء من جميع الأسماء التابعة لدائرة ما .

```
10 READ N$
20 IF N$ = "last" THEN 70
30 PRINT N$
40 DATA "nashid","ahmed","fayez","last"
50 READ N$
60 GOTO 20
70 END
```

ونجد أن البيانات المخزنة داخل N\$ هي كالتالى :

N\$	قائمة
NASHID	NASHID
AHMAD	AHMAD
FAYEZ	FAYEZ
LAST	

وهذا مثال مبسط ، والبحث فى القائمة عن اسم شخص معين ، «فايز» مثلاً ، أمر سهل . ولكن لو كان عدد أسماء الموظفين فى دائرة معينة كبيراً حوالى ٥٠٠ مثلاً فالبحث عن اسم معين يتطلب جهداً كبيراً ، إلا إذا كانت قائمة الأسماء مرتبة حسب الأحرف الهجائية .

وفى معظم الأحيان تكون الأسماء المقروءة عشوائية الترتيب ، ويطلب من الحاسب فرزها ، وإذا أردنا تحقيق ذلك ، يجب أن نخزن الأسماء كلها فى الذاكرة ، ومن ثم يتم فرزها ، وهذا يتطلب مقارنة بين متغيرات حرفية .

إن المتغيرات الحرفية يتم ترميزها داخل الحاسب بأرقام / أرقام وحروف وفق أحد النظامين (الثمانى أو السداسى عشر) (مقارنة بالنظام العشرى الذى يستخدمه الإنسان) . ولا يهمنا هنا التعرض لكيفية تمثيل البيانات وفق هذين النظامين ، بل المهم أن نوضح الأساس الذى تستند إليه المقارنة بين المتغيرات الحرفية .

فمقارنة A مع B أو AB مع AC تستند إلى قيم رقمية لها معنى محدد للحاسب . فالحرف A له رقم أصغر من رقم B وأرقام الأحرف المجمعة AB أصغر من AC... وهكذا .

وأحد الأساليب المقترحة لفرز الأسماء الثلاثة هو أن تخزن الأسماء الثلاثة في صناديق مختلفة N1\$, N2\$, N3\$ ومن ثم تتم المقارنة فيما بينها باستخدام عبارات IF / THEN لفرض فرزها ، وهذا يمكن تحقيقه إذا كان عدد الأسماء ثلاثة أو أربعة . أما إذا زاد العدد على ذلك ، فإن متابعة تسلسل عبارات IF / THEN تكاد تكون مستحيلة على العقل البشرى . وأدى هذا إلى تطوير أسلوب في البرمجة يساعد في التحكم في البيانات التابعة لشيء معين موصوف ، أسماء الموظفين في هذا المثال البسيط ، فبدلاً من استخدام أسماء متغيرات لصناديق مستقلة ، فإنه يمكن الإشارة إلى مجموعة صناديق ذات علاقة وإطلاق اسم متغير واحد عليها ، ومن ثم تعريف كل صندوق برقم معين .

ومن أمثلة الحياة العملية عن هذا الأسلوب :

— نكتب بريد منطقة ما ، وأرقام صناديق البريد التابعة لها .

— اسم فندق معين وأرقام الغرف فيه .

— اسم شارع معين وأرقام المنازل فيه .

— اسم منطقة جغرافية معينة والرمز البريدي لها .

أما قواعد البرمجة الخاصة بهذا الأسلوب فتتم كالتالى :

يعطى اسم لمجموعة الصناديق ، ويشار إلى كل صندوق برقم متسلسل من ١ إلى عدد الصناديق المطلوبة — ١٠ مثلاً — وذلك كالشكل التالى :

شكل المصفوفة ذات البعد الواحد

<div style="display: flex; justify-content: space-between; align-items: center;"> N\$ ← اسم المتغير للمجموعة → رقم الصندوق : X </div>									
1	2	3	4	5	6	7	8	9	10
NASHID	AHMED	FAYEZ	LAST						

و يشار إلى كل صندوق كالتالى :

N\$(x)

حيث x تشير إلى رقم الصندوق ، ويبدأ من ١ ، ٢ ، ٣ ... إلى ١٠ .

ففى المثال السابق ، نود تخزين الأسماء فى الصناديق كما هو موضح سابقاً فمن الممكن تبديل سطر 10 ليظهر كالتالى :

10 READ N\$(1), N\$(2), N\$(3)

ومع إلغاء سطر 20 وتعديل سطر 40 بإلغاء «LAST» من نهايته ، يمكن قراءة الأسماء الثلاثة وتخزينها فى الصناديق الثلاثة الأولى ذات الأرقام ١ ، ٢ ، ٣ من اسم المجموعة N\$.

ولكن ازدياد عدد الأسماء يودى بهذه الطريقة ويجعلها عديمة الاستعمال ، لذلك لابد من وجود وسيلة أخرى . وهذه الوسيلة تتطلب استخدام متغير (عداد) لأرقام الصناديق ومن ثم تغير هذا العدد كلما تم تخزين اسم من البيانات ، وبعد التعديلات عليه يصبح البرنامج كالتالى :

```
5 LET X = 1
10 READ N$(X)
20 IF N$(X) = "last" THEN 70
30 PRINT N$(X)
40 DATA "nashid", "ahmed", "fayez", "last"
45 LET X = X + 1
50 READ N$(X)
60 GOTO 20
70 END
```

استخدام تعليمة DIM : وتتطلب المصفوفة تعليمة معينة تحدد عدد الصناديق التي تريد من الحاسب أن يحجزها لكل اسم . وتتبع هذه التعليمة الشكل التالى :

شكل عبارة DIM			
xxx ↓ رقم السطر	DIM ↓ تعليمة تحديد أبعاد المصفوفة	YY ↓ اسم المتغير	(20) YX , (10) ↓ وهكذا عدد الصناديق

تطوير برنامج باستخدام المصفوفات ذات البعد الواحد :

مثال (١١-١) :

الهدف : حساب الربح الصافى لكل يوم فى الأسبوع لبائع الجرائد ، إذا علم أن سعر بيع الجريدة ريالان وتكلفتها نصف ريال وثمان المسترجع منها ٢٥ هللة (ربع ريال) ، وحساب نسبة مبيعات كل يوم إلى المجموع الكلى للمبيعات .

أولاً - طرق الحل :

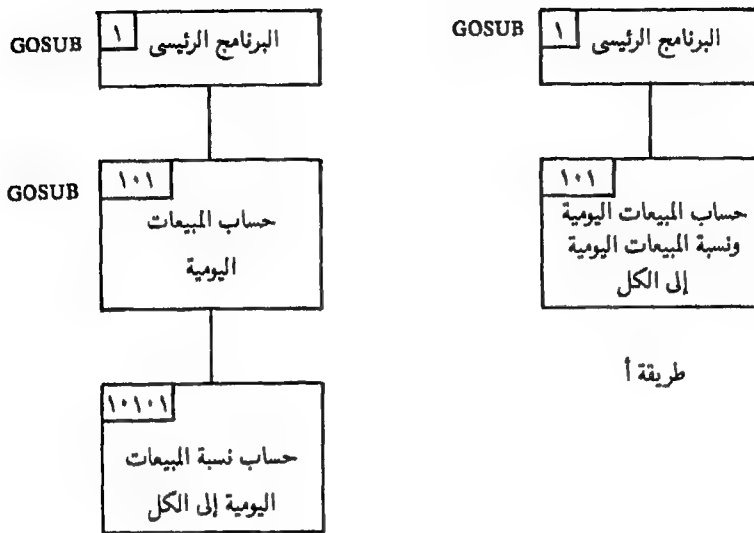
عند تحليل هذا الهدف ، يتبين أنه فى الإمكان تقسيمه إلى جزأين :

- الأول - يقوم بحساب الصافى اليومى والأسبوعى للمبيعات .
- الثانى - يقوم بحساب نسبة المبيعات اليومية إلى المجموع الكلى للمبيعات .

ويمكن اتباع أحد ثلاثة طرق لتحقيق هذا الهدف :

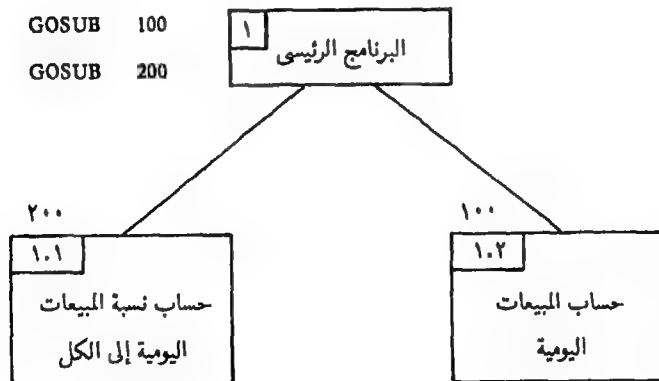
- أ - أن يظهر الجزءان فى برنامج فرعى واحد .
- ب - أن يظهر الجزءان فى برنامجين فرعيين مستقلين ، وفى هذه الحالة :
 - ١ - إما أن يكون التنفيذ من البرنامج الرئيسى لكلا الجزأين بشكل متتال .
 - ٢ - وإما أن يكون تسلسل التنفيذ من البرنامج الرئيسى إلى البرنامج الفرعى الأول ومنه إلى البرنامج الفرعى الثانى .

ثانياً - الهيكل الهرمية حسب هذه الطرق الثلاثة مع تبيان الأجزاء التي ستظهر فيها تعليمة GOSUB للتوضيح .



طريقة أ

طريقة ب - ٢



طريقة ب - ١

ففى الشكل ب - ٢ نجد أن الجزء رقم ١٠١ يتحكم فى الجزء ١٠١٠١ ، وفى نفس الوقت يقوم بالمعالجات الخاصة به ، ففى هذه الحالة يفضل ألا تكون تبعية الجزء ١٠١٠١ له ؛ وذلك للأسباب التالية :

- ١ - أن تعديل الجزء ١٠١ قد يؤدى إلى التأثير على الجزء ١٠١ .
 - ٢ - أن وظيفة الجزء المتحكم الرئيسية هى توجيه التنفيذ للأجزاء الأخرى التابعة له .
 - ٣ - صعوبة تتبع خطوات البرنامج ، وخاصة إذا تكرر هذا النوع من العلاقة بين الجزء المتحكم والمنفذ فى برنامج ما .
- وقد تم استخدام الأسلوب الأول لتطوير البرامج للأمثلة المشروحة فى الفصول السابقة ؛ لذلك فسنبوم بتطوير البرنامج حسب الهيكل الهرمى ب - ١ .

الخطوات الرئيسية فى الجزء الأول :

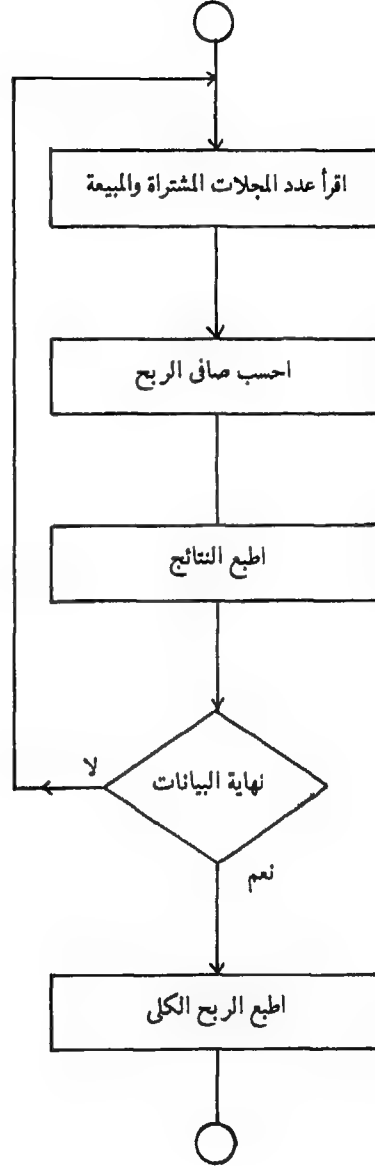
- ١ - لكل يوم من أيام الأسبوع ، اقرأ الكمية المشتراة ، والكمية المباعة .
- ٢ - لكل يوم احسب صافى الربح = الكمية المباعة * 2.0 - الكمية المشتراة * 50 . + (الكمية المشتراة - الكمية المباعة) * 25 .
- ٣ - اطبع الناتج .
- ٤ - احسب المبلغ الإجمالى المتجمع بعد كل يوم .
- ٥ - وبعد انتهاء معالجة اليوم السابع ، اطبع المبالغ الكلية ثم توقف .

خطوات الحل للجزء الثانى :

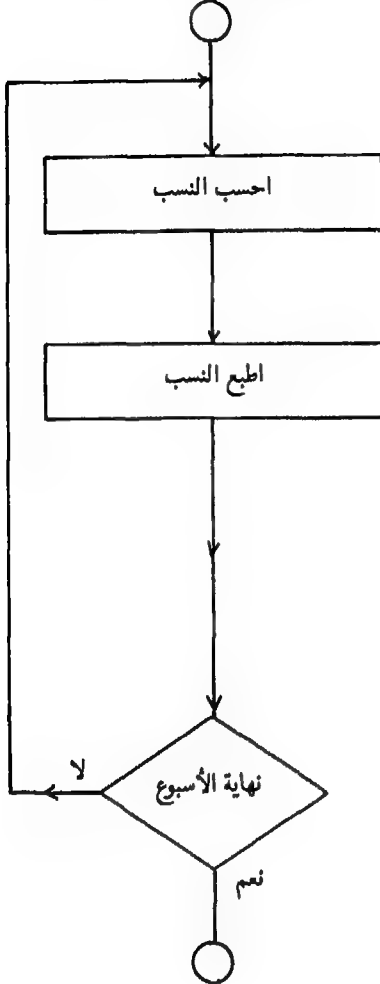
- (١) لكل يوم من أيام الأسبوع ، احسب نسبة الربح إلى المجموع الكلى للربح .
- (٢) اطبع الناتج .
- (٣) بعد طباعة آخر نسبة ، توقف .

ثالثاً - نمط البرمجة التركيبية :

الجزء الأول : حساب الأرباح



الجزء الثاني : حساب النسب



رابعاً - تطوير البرنامج :

وكيفية تحقيق ذلك باستخدام المصفوفات تتم حسب الجدول التالي :

اسم المصفوفة رقم الأسبوع M	P (I) عدد المشتري	S (I) العدد المبيع	R (I) صافي الربح	G (I) النسبة من كل الأرباح
1	50	40	XX.XX	XX.XX%
2	30	15	XX.XX	XX.XX
3	60	45	XX.XX	XX.XX
4	80	60	XX.XX	XX.XX
5	70	40	XX.XX	XX.XX
6	35	15	XX.XX	XX.XX
7	55	45	XX.XX	XX.XX
↑ رقم الصندوق في كل مصفوفة		T →	XXXXXX	المجموع

من الجدول السابق يتبين أننا بحاجة إلى أربع مصفوفات ذات بعد واحد ، ومتغير
عداد لمعرفة أرقام الأسبوع ، ومتغير جامع لمعرفة مجموع المبيعات الكلية . لاحظ أن
العداد M يرمز إلى رقم الصندوق لكل مصفوفة بالإضافة إلى رقم الأسبوع ، ويمثل الشكل
التالي البرنامج الكامل حسب الهيكل الهرمي ب - ١ .

شكل (١١ - ١)

برنامج لحساب صافي أرباح بائع الجرائد أسبوعياً ونسبة ربح كل يوم إلى مجموع الأرباح

```

10 REM
20 GOSUB 50 : REM      استخدام برنامج فرعي لحساب الأرباح و مجموعها
30 GOSUB 180 : REM     استخدام برنامج فرعي لحساب نسبة الأرباح من المجموع
40 END
50 REM      برنامج فرعي لحساب الأرباح
60 LET M = 1
70 REM      لقراءة الأعداد المشتراة و المباعه
80 READ P(M),S(M)
90 REM      حساب صافي الأرباح
100 LET R(M) = S(M) * 21 - P(M) * .5 + (P(M) - S(M)) * .25
110 LET T = T + R(M)
120 LET M = M + 1
130 IF M <= 7 THEN 80
140 DATA 50,40,30,15,80,45,80,60,70,40,35,15,55,45
150 RETURN
160 REM      برنامج فرعي لحساب النسب و طباعة النتائج
170 PRINT "      الرقم      العدد      صافي الربح      النسبة من"
180 PRINT "      المشتري      المباع      المجموع الأرباح"
190 PRINT "===== "
200 LET W = 1
210 LET C(W)=(R(W) / T) * 100
220 LET P# = "    ##    ##    ##.##    ##.##"
230 PRINT USING P#;W,P(W),S(W),R(W),C(W)
240 LET W = W + 1
250 REM      لا تتفسر من نهاية الأسابيع
260 IF W <= 7 THEN 210
270 PRINT "===== "
280 PRINT TAB(42);T;" = مجموع الأرباح"
290 PRINT "===== "
300 RETURN
  
```

وعند تنفيذ البرنامج ، سنحصل على النتائج التالية :

النسبة من مجموع الأرباح	صافي الربح	العدد المباع	العدد المشتري	رقم الأسبوع
=====	=====	=====	=====	=====
15.07	57.50	40	50	1
5.21	18.75	15	30	2
17.71	63.75	45	80	3
23.61	85.00	60	80	4
14.58	52.50	40	70	5
4.86	17.50	15	35	6
18.06	65.00	45	55	7
=====				
مجموع الأرباح = 360				
=====				

لاحظ أن المتغير العداد M و W قد استخدم للدلالة على رقم الأسبوع ، بالإضافة إلى رقم الصندوق في المصفوفة . لاحظ أيضا أن تعليمة DIM لم تظهر في هذا البرنامج ، إذ أن البرنامج يحدد تلقائياً ١٠ صناديق لكل متغير استعمل كمصفوفة . ولكن من الأفضل إضافتها دائماً إلى البرنامج ؛ منعاً للإرباك والمزج بين متغيرات المصفوفات والمتغيرات العادية ، ويتم ذلك في البرنامج السابق كالتالى :

107	DIM	R (10) , S(10), P(10)
505	DIM	C (10)

مقدمة عن الدوارة البسيطة :

تحدثنا في المثال السابق عن استخدام المتغير العداد ؛ للتحكم في عمليات إدخال وطبع البيانات إلى ومن المصفوفات ذات البعد الواحد . ونظراً لتعدد هذا النوع من التطبيقات في المجالين التجارى والحكومى ، كان لابد من تطوير تعليمات برمجية خاصة لتسهيل هذا النوع من المعالجات .
ولاحظنا أن هناك ثلاث حالات للمتغير العداد «I» في المثال السابق ، يجب مراعاتها :

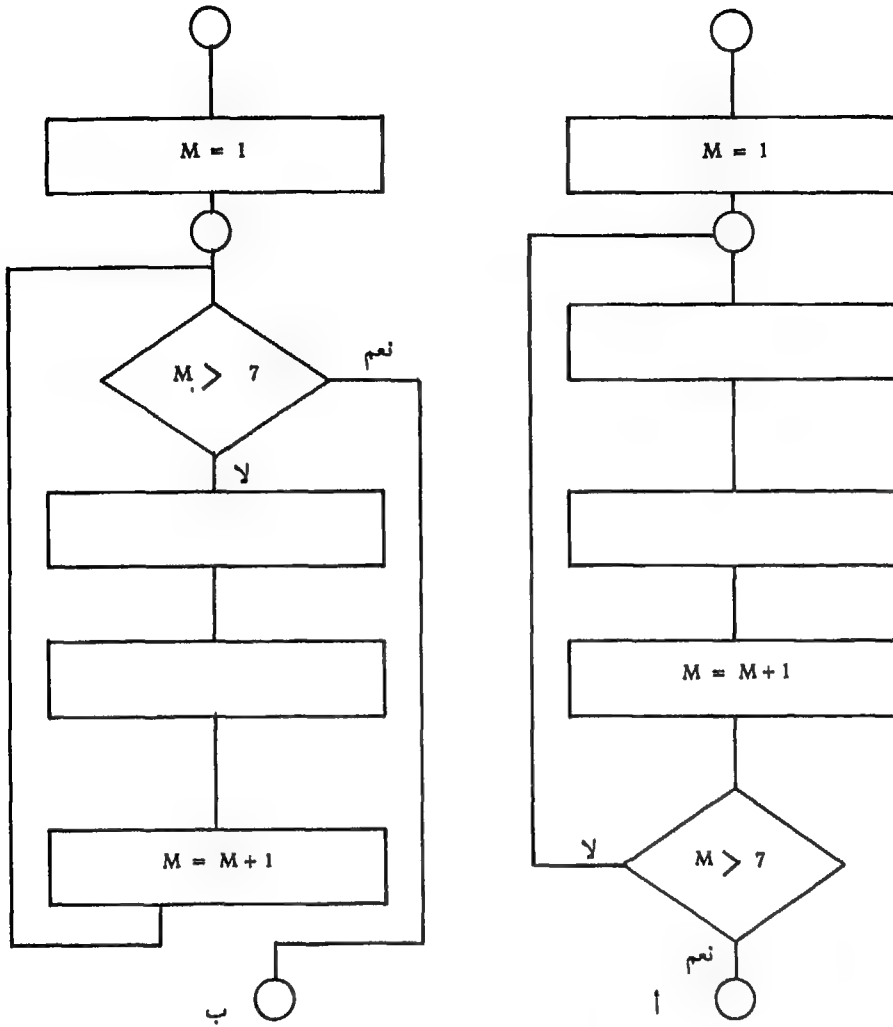
١ — القيمة الأولية ، وكانت ١

٢ — القيمة النهائية ، وكانت ٧

٣ — مقدار الزيادة في العداد وكانت ١

وكان لابد من تحديد هذه القيم الثلاث للعداد المستخدم في التحكم في الدوارة ، سواء كان ذلك عند استخدام المصفوفة أو عند التحكم في تنفيذ مجموعة عبارات بعضها ذات علاقة مع بعض .

و يبين الرسمان التاليان تسلسل العمليات المنفذة للتحكم في الدوارة بأسلوبين مختلفين وفق أساليب البرمجة التركيبية .



وكما مر ذكره في الفصل السابع ، لوحظ أن الرسم أ هو الذي استخدم في البرنامج السابق شكل (١١ - ١) ، وذلك للتقليل من عدد استخدامات تعليمة GO TO . ولكن الرسمين سليمان من وجهة نظر منطقية . وسنتحدث الآن عن التعليمات التي تسهل عملية التحكم في الدوارة البسيطة ، حيث تستخدم هذه التعليمات المنطق الموضح في الشكل - ب - السابق .

تطوير برنامج باستخدام تعليمتى FOR / NEXT للتحكم فى الدوارة البسيطة :
فى المثال التالى ، سيتم تطوير برنامج باستخدام تعليمات الدوارة البسيطة ؛
للتحكم فى تنفيذ عبارات ذات علاقة ، ودون استخدام المصفوفات ، وذلك باستخدام
مثال محاسبى .

مثال (١١ - ٢) :

الهدف : حساب قيمة الاستهلاك للأصول الثابتة بطريقة الاستهلاك المتناقص ، إذا
علمت القيمة الأصلية الدفترية وعدد سنوات الاستخدام ، وذلك بأسلوب مجموع أعداد
السنوات .

أولاً - خطوات الحل :

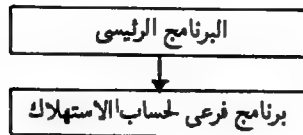
١ - الحصول على القيمة الشرائية وسنوات الاستخدام .

٢ - حساب مجموع أرقام السنوات ، مثال :

إذا كان الاستهلاك لمدة ٥ سنوات ، يكون مجموع أرقام
السنوات : $1+2+3+4+5 = 15$ و يكون مقدار الاستهلاك السنوى للأصل =
القيمة الدفترية ÷ رقم السنة + ١٥ ، وذلك بدءاً بالرقم ٥ للسنة الأولى وانتهاء
بالرقم ١ للسنة الخامسة .

٣ - حساب الاستهلاك السنوى لكل السنوات .

ثانياً - الهيكل الهرمى : يمكن تطوير الهيكل الهرمى بأحد أسلوبين :
(أ) وجود جزء واحد ، وذلك حسب الشكل التالى .



ب - وإذا أريد استخدام أكثر من جزء واحد ، فيمكن الأخذ بعين الاعتبار للوظائف التالية :

١ - الحصول على المدخلات .

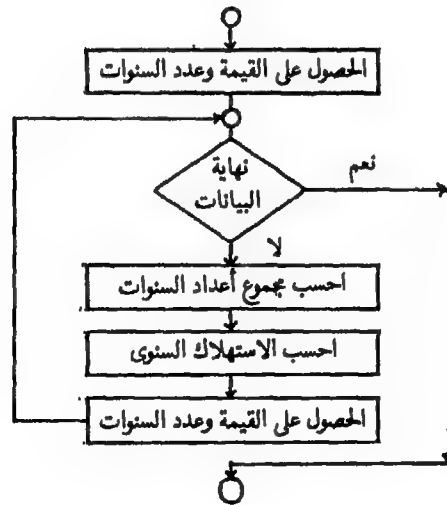
٢ - حساب عدد السنوات .

٣ - حساب الاستهلاك السنوي لكل السنوات مع الطباعة للنتائج .

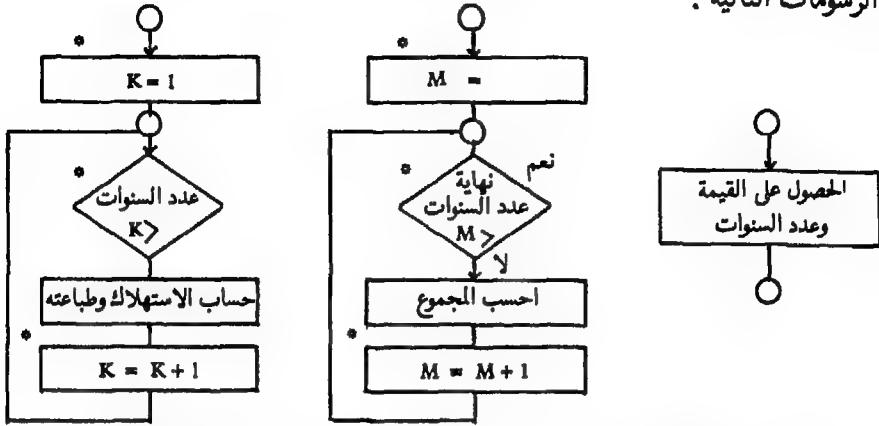
حيث يتم تدقيق نهاية البيانات عن طريق البرنامج الرئيسى ، ويرجع سبب ذلك إلى ضرورة إنهاء البرنامج في الجزء الرئيسى وليس في أى جزء فرعى .
ويعكس الرسم التالى الهيكل الهرمى الذى يحوى ثلاثة أجزاء فرعية .



ثالثاً : رسم البرمجة التركيبية : يعكس الرسم التالى الخطوات المتبعة لتحقيق الهدف من المثال ، على فرض استخدام برنامج جزئى واحد .



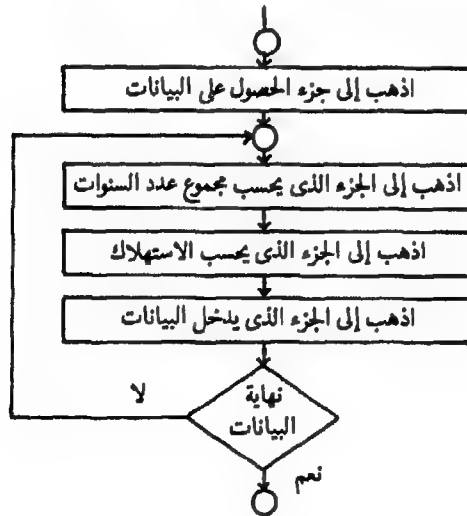
ووفق الأسلوب ب ، يمكن وضع الوصف التفصيلي للخطوات عن طريق الرسومات التالية :



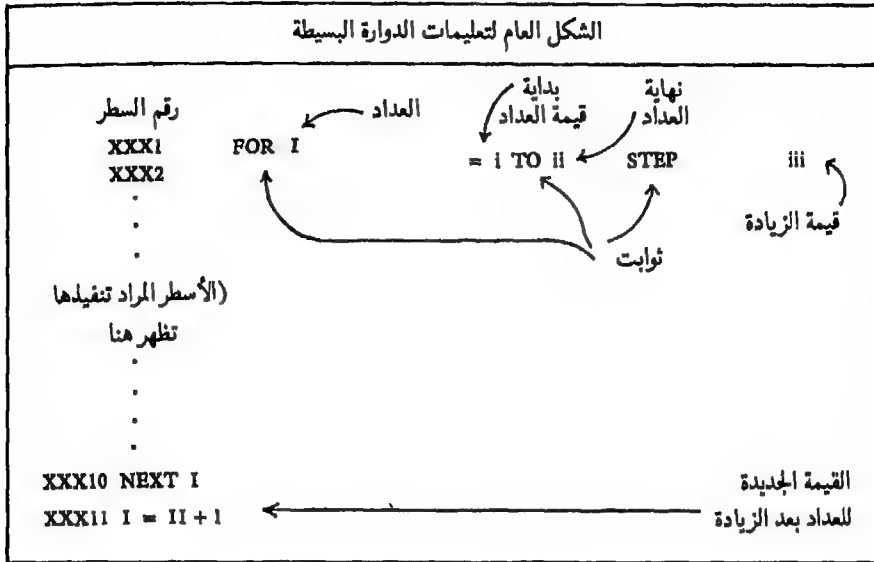
١ - الحصول على المدخلات ٢ - حساب عدد السنوات ٣ - حساب الاستهلاك السنوي

لاحظ الرموز المؤشر عليها ب «*» والتي تشير إلى الخطوات الأساسية للتحكم في الدوارة .

ولاحظ أن البرنامج الرئيسي يتبع الرسم التالي للبرمجة التركيبية :



ويتطلب الجزء الثانى والثالث أسلوباً للتحكم فى الدوارة يمكن تنفيذه باستخدام تعليمتى FOR / NEXT ، كما هو موضح فى الشكل التالى :



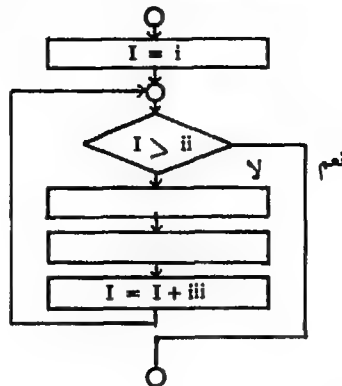
حيث تحصر العبارات المراد التحكم فى تنفيذها بين عبارتى FOR و NEXT .
وتقوم تعليمة FOR بالعمليات التالية :

١ - إسناد القيمة الأولية (بداية العداد) مرة واحدة للمتغير العداد - I الشكل السابق : $I = i$.

٢ - التدقيق فيما إذا كانت قيمة I تزيد على القيمة النهائية للعداد ، فى هذه الحالة $I > ii$ فإذا كانت قيمة العداد تزيد على القيمة المعطاة يتم التشعب إلى السطر الذى يلى عبارة NEXT ، XXX 11 ، وإلا يستمر التنفيذ إلى السطر الذى يلى عبارة FOR ، XXX 2 .

أما تعليمة NEXT فتحسب القيمة الجديدة للمتغير العداد حسب ماتطلبه القيمة المحددة فى عبارة FOR وعلى وجه التحديد القيمة الموجودة بعد كلمة STEP ، iii .

لذلك نرى أن عبارتي FOR / NEXT تتحكمان في الدوارة على الشكل التالي :



ويمكن تتبع ما يحدث لقيمة I وكيفية المقارنة والتشعب حسب الجدول التالي :

التعليمة	قيمة	المقارنة	الحالة	نتيجة المقارنة
FOR	$I = i$	$I > ii$	لا	استمر
NEXT	$I = I + iii$			
FOR		$I > ii$	لا	استمر
NEXT	$I = I + iii$			
FOR		$I > ii$	لا	استمر
NEXT	$I = I + iii$			
FOR		$I > ii$	نعم	توقف

وسنعطى مثالا بسيطاً لمراقبة ما يحدث لقيمة I ، كما في البرنامج التالي :

```

10 FOR I = 1 to 4 STEP 1
20 PRINT I
30 NEXT I
40 PRINT «القيمة خارج الدوارة» ; I
  
```

وعند تنفيذ البرنامج تكون المخرجات كالتالي :

1
2
3
4
5 القيمة خارج الدوارة

رابعاً - البرنامج كاملاً في الشكل التالي :

شكل (١١ - ٢)

برنامج لحساب قيمة الاستهلاك المتناقصة وفق أسلوب عدد أرقام السنوات

```

10 REM برنامج رئيسي ويتم تدقيق نهاية البيانات فيه
20 GOSUB 90 ' للحصول على المدخلات
30 IF V = -999 THEN 80 ' تدقيق نهاية البيانات
40 GOSUB 120 ' عدد السنوات
50 GOSUB 180 ' حساب الاستهلاك السنوي
60 GOSUB 90 ' على المدخلات
70 IF V <> -999 THEN 40 ' تدقيق نهاية البيانات
80 END
90 REM برنامج فرعي لا يدخل قيمة الاصل وعدد السنوات
100 INPUT "أدخل قيمة الاصل وعدد سنوات الاستهلاك أو -٩٩٩ للانهاء"; V, Y
110 RETURN
120 REM برنامج فرعي لحساب مجموع السنوات
130 LET II = 0
140 FOR I = 1 TO Y STEP 1
150 LET II = II + I
160 NEXT I
170 RETURN
180 REM حساب ما استهلك في السنة لكل السنوات
190 FOR I = 1 TO Y STEP 1
200 LET D = V * (6-I) / II
210 PRINT I; " ما استهلك في السنة رقم "; D
220 NEXT I
230 RETURN

```

وفي حالة تنفيذ البرنامج، سنحصل على المخرجات التالية :

2, 1000 ؟ أدخل قيمة الأصل وعدد سنوات الاستهلاك أو -٩٩٩ للانهاء

1 666.6667

2 333.3334

3, 500 ؟ أدخل قيمة الأصل وعدد سنوات الاستهلاك أو -٩٩٩ للانهاء

1 250

2 166.6667

3 83.33334

0, 999 ؟ أدخل قيمة الأصل وعدد سنوات الاستهلاك أو -٩٩٩ للانهاء

OK

ملاحظات على البرنامج :

١ - استخدام العداد I للتحكم في الدوارة في كلا الجزأين ، وهذا لا يؤثر مطلقاً على سير البرنامج ؛ إذ تبدأ القيم الجديدة لها في الجزء الثاني بعد الانتهاء من الجزء الأول .

٢ - عند الانتهاء من تنفيذ الدوارة ، تكون قيمة العداد I أكبر من القيمة النهائية بمقدار الزيادة المحددة عقب STEP BY .

٣ - يحوى السطر 210 أسلوباً مبسطاً للحصول على القيم العكسية للعداد I وذلك حسب الجدول التالى :

قيمة I	القيمة المطلوب استخدامها في المعادلة	كيفية الحصول عليها
1	5	6 - 1
2	4	6 - 2
3	3	6 - 3
4	2	6 - 4
5	1	6 - 5

وهذا يعنى أنه من الممكن استخدام عداد الدوارة في معادلة جبرية داخل الدوارة المنفذة ، شريطة ألا تظهر في الجهة اليسرى لإشارة = ، إذ أن تغير قيمة عداد الدوارة داخل العبارة المنفذة يؤدي إلى تغير عدد المرات التى ستنفذ معها هذه العبارة (المحصورة بين تعليمتى FOR و NEXT) .

ففى المثال التالى ، نجد أن وضع عداد الدوارة إلى يمين إشارة = في معادلة جبرية تقع ضمن العبارات الواقعة بين FOR و NEXT ، أدى إلى تغير عدد المرات التى تنفذ بها هذه العبارة :

تعديل قيمة عداد الدوارة في عبارة بين FOR و NEXT	
FOR / NEXT تعديل قيمة I عداد الدوارة	FOR / NEXT بشكلها الصحيح
10 FOR I = 1 TO 5 STEP BY 1 50 I = I + 2 100 NEXT I	10 FOR I = 1 TO 5 STEP BY 1 100 NEXT I
ب	أ

ففى الشكل — أ — يتم تنفيذ العبارات خمس مرات، أما فى الشكل — ب — فيتم تنفيذ العبارات مرتين فقط، وشرح ذلك كالتالى :

نتيجة المقارنة	الحالة	المقارنة	قيمة	رقم السطر
استمر	لا	$1 > 5$	1	10
		لا توجد مقارنة بعد	3	50
استمر	لا	$4 > 5$	4	100
		لا توجد مقارنة بعد	6	50
توقف	نعم	$7 > 5$	7	100

٤ — التسلسل المتناقص للعداد : فى الفقرة السابقة تم شرح كيفية الحصول على القيم المتنزلة لعداد الدوارة باستخدام قيمها المتصاعدة ، فى حالات كهذه يمكن أن تظهر القيم المتحكممة فى عداد الدوارة بشكل متناقص ، ففى المثال السابق يظهر السطر 150 كالتالى :

150 FOR I = 5 TO 1 STEP — ١

و يتم تعديل السطر 150 ليتماشى مع الشكل الجديد لعبارة FOR

150 LET D = V * I / II

بالإضافة إلى إمكانية استخدام الأرقام السالبة في عبارة FOR ، يمكن استخدام قيم عشرية للمتغيرات فالعبارات التالية كلها صحيحة :

FOR I = 1 TO 10 STEP .1

FOR J = 1 TO 1.5 STEP .01

FOR M = 3.6 TO 0 STEP -.1

٥ — إمكانية استخدام متغيرات بدلاً من قيم في عبارة FOR فالمتغير Y والذي يمثل عدد السنوات ، استخدم في السطرين 150, 200 للدلالة على القيمة النهائية المطلوبة لعدد الدوارة . وكذلك يمكن أن يكون أى من القيم أو جميعها ممثلة في متغيرات ، بشرط إسناد قيم لها قبل تنفيذ عبارة FOR .

فإذا أردنا مثلاً تحديد بداية السنة المطلوب فيها الاستهلاك ، يمكن إدخال هذه القيمة مع بقية المدخلات ، ويتم تغيير البرنامج كالتالى :

V, Y, Y1 ; «أدخل القيمة للأصل مع سنوات الاستهلاك وبداية السنوات أو 999
للانتهاء» INPUT 110

150 FOR I = Y1 TO Y STEP 1

20 FOR I = Y1 TO Y STEP 1

وكذلك الأمر إذا كانت الزيادة لأكثر من سنة واحدة ، أى حساب قيمة الاستهلاك كل سنتين ، فإذا أضفنا المنفذ Y2 في نهاية السطر 110 فتكون عبارة FOR كالتالى :

110 FOR I = Y1 TO Y STEP Y2

٦ — استخدام تعابير جبرية للتحكم في قيم عداد الدوارة :
إذا أردنا حساب الاستهلاك نصف السنوى عوضاً عن الاستهلاك السنوى

للأصل ، تصبح الأرقام الخاصة بالحساب ضعف القيمة المدخلة للسنوات ، أى ١٠ بدلاً من ٥ مثلاً .

ويمكن تعديل عبارتى FOR لحساب الاستهلاك نصف السنوى كالتالى :

```
150 FOR I = 1 TO Y * 2 STEP 1
```

```
200 FOR I = 1 TO Y * 2 STEP 1
```

وكذلك من الممكن أن تحمل العبارات الجبرية محل أى من المتغيرات أو القيم العددية فى عبارة FOR فالأشكال التالية لعبارة FOR صحيحة :

```
FOR I = (I1 / K1) TO W2 * 3 STEP Y / Z
```

```
FOR I = (I1 + I2) / I3 TO I4 * I5 / I6 STEP
```

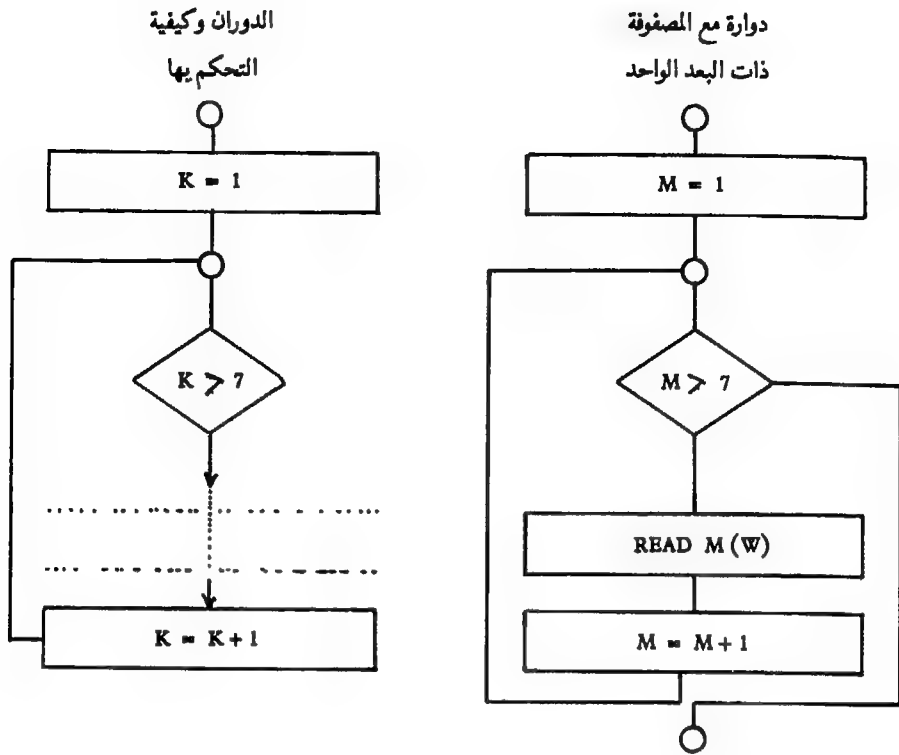
```
(I6 + I7) / I8
```

```
FOR I = 1 TO Y * Y2 / 2 STEP Y3 / 3
```

تطوير برنامج باستخدام FOR / NEXT

لمعالجة المصفوفات ذات البعد الواحد :

تبين مما سبق أن عبارتى FOR / NEXT تسهلان عملية التحكم فى معالجة مجموعة من العبارات ذات العلاقة فيما بينها ، وكيف أن التخزين فى المصفوفة ذات البعد الواحد يتطلب استخدام رقم متسلسل للصندوق ، وأن التحكم فى هذا الرقم يخضع لنفس الأسس التى يخضع لها التحكم فى الدوارة ، ويوضح ذلك الشكلان التاليان :



فعند استخدام تعليمتي FOR / NEXT مع المصفوفة ذات البعد الواحد ، يتم في العادة استخدام عداد الدوارة كرقم للصندوق الذي يظهر بين قوسين بعد اسم المصفوفة .

ففي المثال الأول في هذا الفصل ، يتم تعديل البرنامج الموضح في شكل ١ عند استخدام FOR / NEXT كالتالي :

الأسطر التالية تتحكم في الدوارة الأولى والثانية :

```

105 LET M = 1
140 LET M = M + 1
150 IF M <= 7 THEN 110
  
```

500 LET W = 1 برنامج فرعى لحساب النسب وطباعة النتائج

525 LET W = W + 1

580 IF W <= 7 THEN 510

FOR / NEXT يتم استبدال الأسطر التالية بعبارات

105 FOR M = 1 TO 7

150 NEXT M

500 FOR W = 1 TO 7 برنامج فرعى لحساب النسب وطباعة النتائج

530 NEXT W

ومع إزالة السطرين 525, 140 يصبح البرنامج كاملاً وصحيحاً و يؤدي نفس النتائج . وتنطبق هنا نفس الملاحظة التي ظهرت عقب شكل ١١ - ١ مع الفارق أن عدد الدوارة هو الذى استخدم كرقم الأسبوع ورقم الصندوق ، فى المصفوفات المختلفة بدلاً من أن يكون المتغير العدد .

وسنتعرض الآن لتطوير برنامج آخر باستخدام FOR / NEXT والمصفوفة ذات البعد الواحد .

مثال (١١ - ٣) :

الهدف : قراءة أعمار الموظفين وتقسيمها حسب الفئات التالية ، ومن ثم طبع هذه الأعمار الثابتة لكل فئة مع عددها ونسبة كل فئة :

الفئة الأولى : أقل من ٢٠ سنة .

الفئة الثانية : ٢٠ إلى أقل من ٣٠ سنة .

الفئة الثالثة : ٣٠ إلى أقل من ٤٠ سنة .

الفئة الرابعة : ٤٠ فأكثر .

أولاً - خطوات الحل :

- أ (الحصول على العمر حتى نهاية البيانات .
 ب (معرفة الفئة الخاصة به وتخزين العمر في مصفوفة .
 ج (عند الانتهاء من التصنيف ، طباعة الفئات وإلى جانبها الأعمار .
 وبحيث يكون شكل المخرج كالتالى :

الأعمار	النسبة	العدد	الفئة
xxxx	xx/xx	xx	أقل من ٢٠ سنة
xxxx	xx/xx	xx	٢٠ سنة إلى أقل من ٣٠
xxxx	xx/xx	xx	٣٠ سنة إلى أقل من ٤٠
xxxx	xx/xx	xx	٤٠ سنة فأكثر

وإذا افترضنا وجود الأعمار التالية كمثال :

٣٩ ، ٤٩ ، ١٨ ، ٤٠ ، ٢٠ ، ٣٠ ، ٣١ ، ٤٥ ، ٢٢ ، ١٩

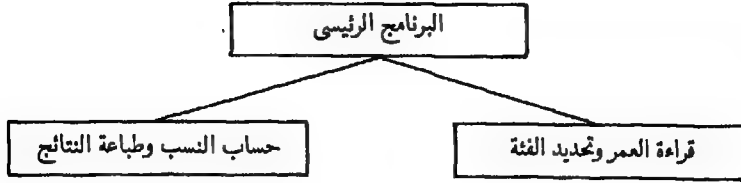
فسيكون شكل المخرج كالتالى :

الأعمار	النسبة	العدد	الفئة
19 18	20%	2	أقل من ٢٠ سنة
22 20	20%	2	٢٠ سنة إلى أقل من ٣٠
31 30 39	30%	3	٣٠ سنة إلى أقل من ٤٠
45 40 49	30%	3	٤٠ سنة فأكثر
	100%	10	المجموع

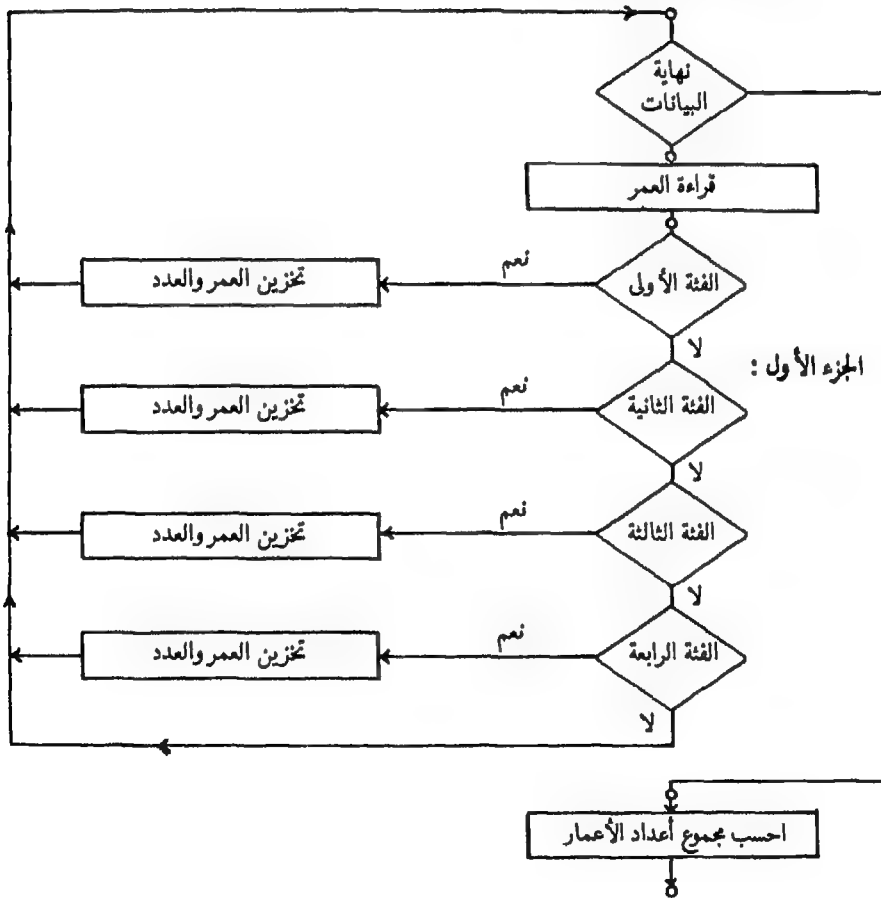
ثانياً - رسم الهيكل الهرمى :

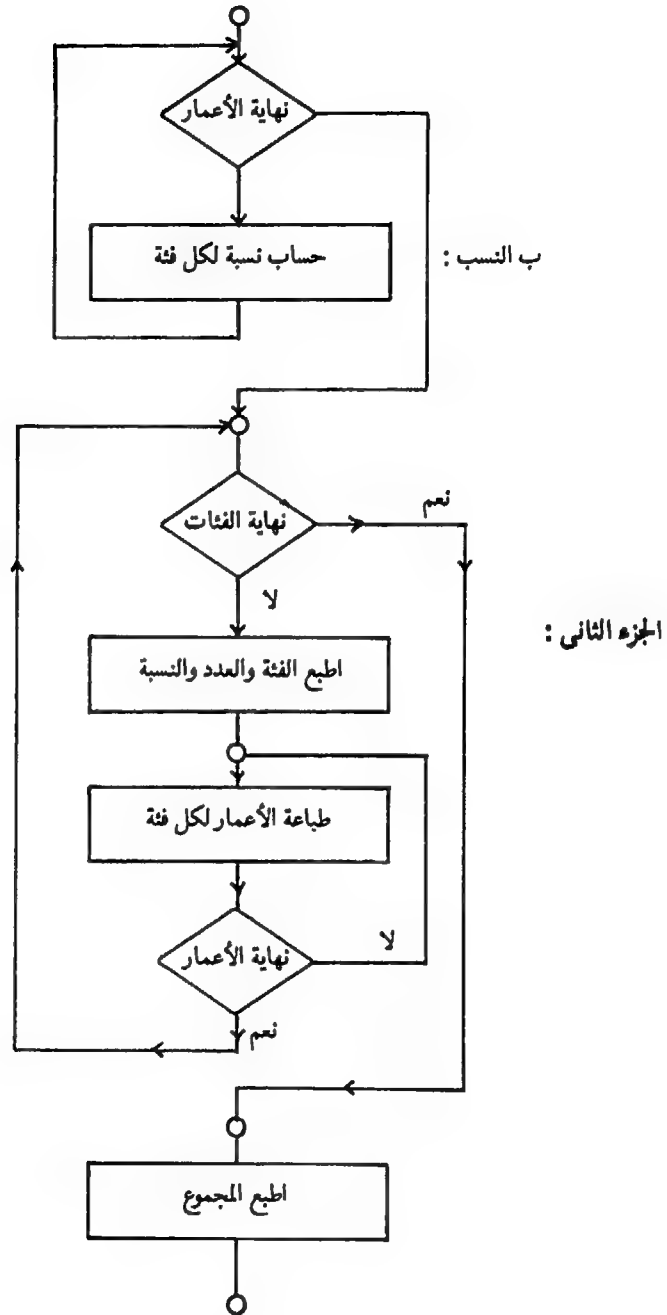
- يمكن أن يقسم البرنامج إلى الأجزاء التالية :
 أ (قراءة العمر وتحديد الفئة وتخزين العمر .
 ب (حساب النسب وطباعة المخرجات .

ويعكس الرسم التالى الهيكل الهرمى لهذا المثال :



ثالثاً - أشكال البرمجة التركيبية :





ملاحظات على أشكال البرمجة التركيبية : نلاحظ من الأشكال السابقة ، التالى :

١ - فى الجزء الأول ، نجد أن مصفوفة لكل فئة قد استخدمت وذلك لتخزين الأعمار فى كل منها ، كما نلاحظ وجود عداد متغير لكل فئة فى مصفوفة واحدة .

٢ - فى الجزء الثانى ، نجد أن طباعة المخرجات تمت عن طريق تنفيذ دوارتين متداخلتين : الأولى لطباعة الفئات الأربع واحدة واحدة ، والثانية لطباعة الأعمار التابعة لكل فئة ، وسيتم التحكم فى الدوارة الأولى بالطريقة العادية وفى الثانية عن طريق FOR / NEXT .

٣ - يكون مجموع المصفوفات المطلوبة كالتالى مطبقاً على مثال الأعمار المشار إليه سابقاً :

NA (I)

1	2	3	4
2	2	3	3

مصفوفة عدد الأعمار فى كل فئة :

PA I

1	2	3	4
20	20	30	30

مصفوفة نسبة الأعمار فى كل فئة

G 1 (I)

19	18
----	----

G 2 (I)

1	2
22	20

مصفوفة لكل فئة تخزن فيها

الأعمار التابعة لها :

G 3 (I)

1	2	3
31	30	39

G 4 (I)

1	2	3
45	40	49

رابعاً - البرنامج كاملاً في الشكل التالي :

شكل (١١ - ٣)

```

برنامج لتصنيف الأعمار إلى فئات أربع، وطباعة أعمار كل فئة مع عددها ونسبتها
10 REM برنامج رئيسي لقراءة الأعمار و تصنيفها و تعدادها
20 DIM NA(4),PA(4),G1(10),G2(10),G3(10),G4(10)
30 GOSUB 100: REM استخدام برنامج فرعي لقراءة الأعمار و تصنيفها و تعدادها
40 GOSUB 500: REM استخدام برنامج فرعي لحساب النسب و طباعة النتائج
50 END
100 REM برنامج فرعي لقراءة الأعمار و تصنيفها و تعدادها
105 REM اسناد القيم الا ولبه للمتغيرات
107 LET C1=0: C2=0: C3=0: C4=0
109 REM دواره بعداد القيم
110 FOR I = 1 TO 10
120 READ A: REM لقراءة العمر
125 REM لتصنيف الأعمار حسب قيمها
127 REM الصنف الاول
130 IF A < 20 THEN C1 = C1 + 1: G1(C1)=A:NA(1)=C1
135 REM الصنف الثاني
140 IF A >= 20 AND A < 30 THEN C2=C2+1: G2(C2)=A:NA(2)=C2
145 REM الصنف الثالث
150 IF A >= 30 AND A < 40 THEN C3=C3+1: G3(C3)=A:NA(3)=C3
155 REM الصنف الرابع
160 IF A >= 40 THEN C4=C4+1: G4(C4)=A:NA(4)=C4
170 NEXT I
180 LET T = C1 + C2 + C3 + C4
185 DATA 13,24,42,64,31,19,20,11,27,38
190 RETURN
500 REM لحساب النسب و طباعة النتائج
505 REM دواره بعداد النسب
510 FOR K=1 TO 4
520 LET PA(K) = (NA(K) / T) * 100 : REM لاجهاد النسبه المئوية
530 NEXT K
540 REM لطباعة النتائج باستخدام دوارتين متداخلتين
550 REM اسناد المتغيرات لمتغيرات
560 LET A1$(1) = " ٢٠ سنة و اقل من ٣٠ سنة " : LET A1$(2) = " ٣٠ سنة و اقل من ٤٠ سنة "
570 LET A1$(3) = " ٤٠ سنة و اقل من ٥٠ سنة " : LET A1$(4) = " اكثر من ٥٠ سنة "
580 REM بدء الطباعة
590 LET I=1
600 IF I > 4 THEN 710
610 PRINT " % ";PA(I); " = النسبة " ;
675 PRINT " (" ;NA(I); " ) " ;A1$(I)
680 PRINT " "
690 LET I = I + 1
700 GOTO 600
710 RETURN

```

وفي حالة تنفيذ البرنامج شكل (١١ - ٣) سنحصل على النتائج التالية :

النسبة = 30 %	(3)	اهل من ٢٠ سنة
النسبة = 30 %	(3)	٢٠ سنة و اقل من ٣٠
النسبة = 20 %	(2)	٣٠ سنة و اقل من ٤٠
النسبة = 20 %	(2)	اكثر من ٤٠ سنة

ملاحظات على البرنامج :

- ١ - لاحظ وجود ؛ في نهاية تعليمة PRINT وهذا يؤدي إلى استمرار الكتابة على نفس السطر عند تنفيذ تعليمة PRINT التالية . لذلك كانت الحاجة إلى سطر 670 لكي نبدأ بالطباعة على سطر جديد عند الانتهاء من كتابة الأعمار لكل فئة ، وإلا بدأ بطباعة الكلمات الواصفة لكل فئة عقب الانتهاء من طباعة الأعمار ، ويمكنك إزالة السطر 670 ولتنفيذ البرنامج تراقب التغير في طباعة النتائج .
- ٢ - استخدام المصفوفة (I) NA في عبارة FOR السطر 610 . وهذا جائز أيضاً ، وفي هذه الحالة ، تكون القيم إما ٢ ، ٢ ، ٣ أو ٣ .
- ٣ - لاحظ عدد المرات التي ستنفذها الأسطر التالية وفق البيانات المعطاة :

السطر	عدد المرات
610	٢
630	٢
640	٣
650	٣

- ٤ - تنفيذ الدوارة الداخلية والدوارة الخارجية : لاحظ أن الدوارة الخارجية ستنفذ الأسطر من 590 إلى 690 أربع مرات ، ولكل مرة من هذه المرات سيتم تنفيذ الأسطر 620 ، 630 ، 640 أو 650 ، بعدد الأعمار لكل فئة كما هو مبين سابقاً .
و يوضح ذلك الجدول التالي :

قيمة I	قيمة K	NA (I)
1	1 , 2	2
2	1 , 2	2
3	1 , 2 , 3	3
4	1 , 2 , 3	3

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ - بين الأخطاء إن وجدت في كل من العبارات التالية :

- 10 FOR I = 1 TO 10 — أ
10 FOR I = 5 TO 1 — ب
10 NEXT — ج
10 FOR R = R TO N — د
10 FOR J = I TO N STEP 4 — هـ

٣ - اعمل على تعديل البرامج التالية لكي يتم تنفيذها بدون أخطاء :

- 10 FOR I = 1 TO 5 — أ
20 PRINT USING A\$; I, I * I
30 NEXT I
40 END

10 READ A, B, C — ب
20 FOR I = B TO C STEP 1
30 PRINT I, A, B, C
40 NEXT I
50 DATA 20, -10, 4, 5, 8, 6
60 END

```

10  READ      N
20  FOR       I = 1 TO N
30  READ      A (I)
40  PRINT     A (I)
50  NEXT      J
60  DATA     9, 4, 2, -4, 6, 8, 7, 3, 9

```

جـ —

٤ — ماهى نتائج المعالجة (المخرجات) التى سنحصل عليها بعد تنفيذ البرامج التالية :

```

10  FOR       I = 1 TO 4
20  LET       A = I * 3
30  PRINT     I, A, I + A
40  NEXT      I
50  END

```

أ —

```

10  LET       A = 10
20  LET       B = 2
30  FOR       I = 1 TO A STEP B
40  PRINT     I, A * B + I
50  NEXT      I
60  END

```

بـ —

```

10  LET       A = 2
20  FOR       I = 10 TO 3 STEP A - 4
30  PRINT     I ;
40  NEXT      I
50  END

```

جـ —

٥ - اعمل على تعديل السؤال الخامس في الفصل العاشر مستخدماً تعليمة NEXT /
FOR للحصول على إجمالى الدخل لجميع البائعين من كلا الصنفين ، وطباعته في
نهاية التقرير .

٦ - اعمل على تطوير برنامج لقراءة رقم القطعة والكمية المخزونة والكمية المطلوبة في
ثلاث مصفوفات أحادية البعد ، ومن ثم إجراء العمليات التالية :

- ١ - حساب الرصيد الجديد وتخزينه في مصفوفة ذات بعد واحد .
- ٢ - طباعة قائمة بأرقام القطع وأرصدها من الكميات ذات الرصيد السلبى ،
وأخرى بالقطع ذات الرصيد الإيجابى وذلك كالتالى :

رقم القطعة	الرصيد	رقم القطعة	الرصيد
1123	15	3211	- 100
2114	10	0011	- 125
3215	20	1023	- 64
.	.	.	.
.	.	.	.
.	.	.	.

٣ - أوجد أدنى رصيد وأعلى رصيد في كل قائمة .

ملاحظة : يمكن تخزين الأرصدة السالبة والأرصدة الموجبة في مصفوفتين منفصلتين
قبل طباعتهما .

أمثلة من البيانات المدخلة :

1123 , 65 , 50

2114 , 70 , 60

3211 , 200 , 300

0011 , 175 , 350

3315 , 60 , 40

المصفوفات ذات البعدين ومكوناتها وتعليمات الدائرة المتقدمة

مقدمة عن المصفوفات ذات البعدين :

تعرف أحادية البيانات المخزنة في جدول معين عن طريق ارتباطها بخاصيتين ،
تحدد الأولى السطر والثانية العمود ، فالجدول التالي يقسم الدارسين للتحقيق بدورة
معينة في معهد الإدارة العامة موزعين حسب خاصيتين : المناطق الجغرافية التي قدموا
منها ، وتقديراتهم الفصلية .

المجموع	المنطقة الجغرافية			التقدير
	الغربية	الشرقية	الوسطى	
١٠	٢	٣	٥	ممتاز
١١	١	٤	٦	جيد جداً
٩	٢	٣	٤	جيد
٦	١	١	٤	مقبول
٤	١	١	٢	راسب
٤٠	٧	١٢	٢١	المجموع

فتحدد أحادية البيانات في الجدول السابق بتقاطع أحد الأسطر وأحد الأعمدة، فالرقم ٦ مثلاً ينتج عن تقاطع السطر الممثل للتقدير «جيد جداً» والعمود الممثل للمنطقة «الوسطى» ، فنقول إن هناك ٦ دارسين من المنطقة الوسطى ذوى تقدير جيد جداً.... وهكذا لبقية الأرقام .

كيفية الإشارة إلى البيانات : كما مر ذكره في الفصل السابق ، فإن أرقاماً لابد أن تستعمل للإشارة إلى محتويات الجدول من البيانات . ففي المصفوفة ذات البعد الواحد ، تم استخدام رقم واحد فقط ، أما في المصفوفة ذات البعدين فلا بد من استخدام رقمين منفصلين ، الأول للسطر والآخر للعمود . ويشبه نظام أرقام الغرف في فندق يحتوى على أكثر من طابق واحد نظام الأرقام الذى نحن بصدد شرحه ، فالرقم «٣١٤» مثلاً يرمز إلى الطابق الثالث ، حجرة رقم ١٤ . والفرق بين أرقام الحجرات وأرقام المصفوفة المعنية ، أن رقمى الأخيرة يفرق بينهما بالفاصلة « ، » ، فالرقم (١ ، ٤) يرمز إلى البيانات المخزنة في الصندوق الناشئ عن تقاطع السطر الأول والعمود الرابع ، ويكون كل عدد الصناديق القابلة لتخزين البيانات يساوى عدد الأعمدة مضروباً x عدد الأسطر .

ونفرق هنا بين أرقام الصناديق ومحتوياتها من البيانات ، تماماً كما في المصفوفة الواحدة . فالصندوق ذو الرقم (٢ ، ٢) في المثال السابق يحتوى على ٤ دارسين من المنطقة الشرقية ذوى تقدير جيد جداً .

و يبين الشكل التالى أرقام الصناديق بالإضافة إلى محتوياتها من البيانات كما في المثال السابق .

رقم العمود ٢		رقم السطر
٢,١		١
٢,٢		٢
٢,٣		٣
٢,٤		٤
٢,٥		٥
٢,٦		٦

وكذلك أرقام الصناديق في المصفوفة المرتبة قطرياً في منتصفها باتجاه تسلسل الأرقام ، نجد أنها متماثلة في رقم السطر ورقم العمود ، كما هو موضح في الشكل التالي :

		١,١
	٢,٢	
٣,٣		

وذلك كله حسب الترتيب الظاهر في الجدول . ففي مصفوفة «الدارس» يشير الصندوق رقم (٢,٣) إلى عدد الدارسين من المنطقة الشرقية ذوى التقدير «جيد» ، أو على العكس من ذلك ، عدد الدارسين الذين حصلوا على تقدير «جيد» وهم من المنطقة الوسطى . ونلاحظ هنا أن رقم السطريأتى أولاً متبوعاً بفاصلة ثم رقم العمود . ويمكننا أن نمثل رقم المنطقة أفقياً والتقدير عمودياً ، أى تدوير الجدول بزاوية ٩٠° . فيكون للجدول أربعة أسطر وستة أعمدة . ولا بد من تغيير أرقام الصناديق تبعاً لذلك .

العلاقة بين أرقام الأسطر والأعمدة :

إن معرفة كيفية تغيير أرقام الأعمدة والأسطر في المصفوفة تؤدي إلى تسهيل برمجة المعالجات التى ستخضع لها البيانات المخزنة فيها . فلو أخذنا سطرأ أفقياً معيناً ، لوجدنا أن رقم السطر ثابت (الرقم الأول) فى حين يتغير رقم العمود (الرقم الثانى) كما هو موضح فى الشكل التالى :

← رقم العمود			
٤	٣	٢	١
٤,١	٣,١	٢,١	١,١
← رقم السطر			

ولو أخذنا عموداً معيناً ، لوجدنا أن أسلوب التغيير على العكس من ذلك ، فالرقم الأول — رقم السطر — يتغير فى حين يبقى الرقم الثانى — رقم العمود — ثابتاً ، كما

← رقم العمود		← رقم السطر	
١,١		١	
١,٢		٢	
١,٣		٣	
١,٤		٤	

وفي حالة تساوى عدد الأسطر مع عدد الأعمدة ، فإن هذا الخط القطرى من الصناديق يقسم المصفوفة إلى قسمين متساويين فى عدد الصناديق .

أما أرقام الصناديق فى القسمين المتساويين ، فلتغيرها نظام خاص بها أيضا . ففى الشكل التالى :

٣,١	٢,١	١,١
△	*	
٣,٢	٢,٢	١,٢
□		*
٣,٣	٢,٣	١,٣
	□	△

نجد أن كل صندوقين متقابلين فى موقعيهما يحدث تبادل بين رقمى سطريهما وعموديهما ، كما هو مـشار إليه بالرموز * ، △ ، □

المجاميع الأفقية والعمودية :

قد يتطلب الأمر فى كثير من التطبيقات إيجاد مجاميع الأعمدة ، والأسطر ، وللحصول على هذه المجاميع ، نتبع أيضاً نظاماً خاصاً . فللعמוד التالى رقم ٣ ، يوضع المجموع فى السطر الرابع :

٣	٣,١
	٣,٢
	٣,٣
	٣,٤
المجموع ←	

وتكون المعادلة لإيجاد مجموع القيم المخزنة في الصناديق التابعة لهذا العمود :

$$(3,3) + (3,2) + (3,1) = (3,4)$$

ونلاحظ تغير أرقام الأعمدة والأسطر على النحو التالي :

أ) عدم تغيير رقم العمود .

ب) زيادة سطر صندوق المجموع ب «١» عن آخر رقم للأسطر .

و يكون تغيير الأرقام على العكس من ذلك عند إيجاد المجموع لسطر معين ، كما في

السطر التالي رقم ٢

٤	٣	٢	١	
٤,٢	٣,٢	٢,٢	١,٢	٢

↑ المجموع

وهكذا يكون تغيير أرقام صناديق مجاميع الأعمدة والأسطر في أية مصفوفة ، ومن الممكن مراجعة الشكل ١٢ - ١ في بداية هذا الفصل للتدقيق في كيفية تغيير أرقام صناديق التخزين في كل سطر وعمود .

مقدمة عن الدوارة المركبة وكيفية استخدامها للتحكم في عمليات المصفوفات ذات البعدين :

لاحظنا في الفصل السابق كيف سهلت تعليمات الدوارة البسيطة معالجة البيانات المخزنة في المصفوفة ذات البعد الواحد . وتم ذلك عن طريق استخدام عداد الدوارة للدلالة على رقم الصندوق التابع لمصفوفة معينة ، حيث يوضع هذا الرقم بين قوسين عقب اسم المصفوفة ، كالتالي $N(I)$ حيث يرمز I إلى رقم الصندوق الذي يتغير من ١ إلى عدد الصناديق الكلي ، وهو نفسه عداد الدوارة .

أما في حالة المصفوفة ذات البعدين ، فنحن بحاجة هنا إلى رقمين للدلالة على صندوق معين ، الأول للدلالة على رقم السطر ، والثاني للدلالة على رقم العمود ، وتتم الإشارة إلى صندوق معين في الدائرة ذات البعدين عن طريق وضع الرقمين — العمود والسطر — بين قوسين ، ومفصولين بالفاصلة « ، » ، كالتالي :

(1,2) S للدلالة على الصندوق في السطر الأول والعمود الثاني في المصفوفة S . وكذلك يمكن استخدام متغيرات وإسناد قيم لها كالتالي :

S (I,J) حيث يتغير I من ١ إلى عدد الأسطر و J من ١ إلى عدد الأعمدة .

وكما كان الشأن مع الدائرة البسيطة ، فإن الدائرة المركبة يمكن استخدامها في حالات تستدعي معالجة بيانات دون تخزينها في مصفوفة ذات بعدين ، وإن كان هذا الأخير يمثل أشهر وأهم تطبيق لها .

وقد مررنا في الفصل السابق كيف يمكن أن تتوالى الدورات البسيطة في برنامج واحد ، وذلك على الشكل التالي :

```
[ 100 FOR I 1 TO 10
  150 NEXT I
```

```
[ 160 FOR J = 1 TO 20 STEP 2
  200 NEXT J
```

.....

وهكذا

فماذا يحدث لو وضعنا دائرة بسيطة داخل أخرى ؟ وكيف تكون العلاقة بين عددي الدورتين ؟

لاحظ أن الشكل التالي للتداخل بين الدورات غير مقبول حسب قواعد البرمجة :

```

100 FOR J = 1 TO 3
  120 FOR K = 1 TO 5 STEP 2
  150 NEXT J
250 NEXT K

```

لذلك فإن الشكل الصحيح للدورة المركبة أن تحتوى كل واحدة منها الأخرى احتواء تاما، ومن الممكن تداخل أكثر من دورتين، كما في المثالين التاليين :

<pre> 100 FOR W = 1 TO 3 200 FOR Z = 1 TO 5 400 NEXT Z 500 NEXT W </pre>	<pre> 100 FOR W = 1 TO 3 200 FOR K = 1 TO 2 300 FOR M = 1 TO 5 NEXT M NEXT K NEXT W </pre>
--	--

وترمز النقاط الظاهرة بين العبارات إلى إمكانية وجود عبارات لغة البيسك أو عدمه، أى أن عبارات PRINT، READ، LET... إلخ قد تتناثر، حسب طبيعة البرنامج، بين عبارات الدورة المركبة. مع الحرص على عدم التشعب من داخل إحدى الدورات إلى آخرها، وإنما يكون التشعب إلى بداية الدورة أو إلى نهايتها حسب منطق البرنامج.

تغير قيم عدادات الدوارة : وحتى نفهم كيفية استخدام الدوارة المركبة لابد من معرفة العلاقة بين القيم التي تأخذها الدوارة، ولنأخذ المثال التالي :

```

100  FOR   I = 1 TO 3
.....
150  FOR   J = 1 TO 2
.....
200  NEXT  J
.....
250  NEXT  I

```

إن قيم I للدورة الكاملة هي كالتالي : ٣ ، ٢ ، ١

وقيم J للدورة الكاملة هي كالتالي : ٢ ، ١

و يتم تتابع قيمة I و J كالتالي :

لكل قيمة واحدة من قيم I، تتم J دورتها الكاملة ابتداء من أول قيمة لها «١» وانتهاء بآخر قيمة لها «٢»، و يوضح ذلك الشكل التالي :

	قيمة J	قيمة I
الدورة الأولى	١	١
	٢	١
الدورة الثانية	١	٢
	٢	٢
الدورة الثالثة	١	٣
	٢	٣

لاحظ :

- (١) أن عدد الدورات الكاملة للدائرة المركبة تتحكم فيها قيم عداد الدائرة الخارجية.
- (٢) أن عدد المرات التي يتم فيها تنفيذ العبارات الظاهرة بين عبارتي FOR و NEXT في الدائرة الداخلية هي ناتج ضرب القيمتين النهائيتين لعددي الدائرة الداخلية والدائرة الخارجية $= 3 \times 2 = 6$ في هذه الحالة.
- (٣) أن عدد المرات التي يتم فيها تنفيذ العبارات الواردة بين FOR الأولى و FOR الثانية يساوي ٣ — القيمة النهائية لعداد الدائرة الخارجية.
- (٤) أن قيمة عداد الدائرة الداخلية «J» تساوي ٣ عند انتقال التنفيذ من عبارة NEXT J إلى عبارة NEXT I.
- (٥) بعد الانتهاء من تنفيذ NEXT I للمرة الأخيرة، تكون $I = 4$ و $J = 3$ وذلك بزيادة «١» على القيمة النهائية المعلنة في عبارتي FOR، أو بزيادة الرقم الذي يظهر بعد STEP في حالة اختلافه عن «١».

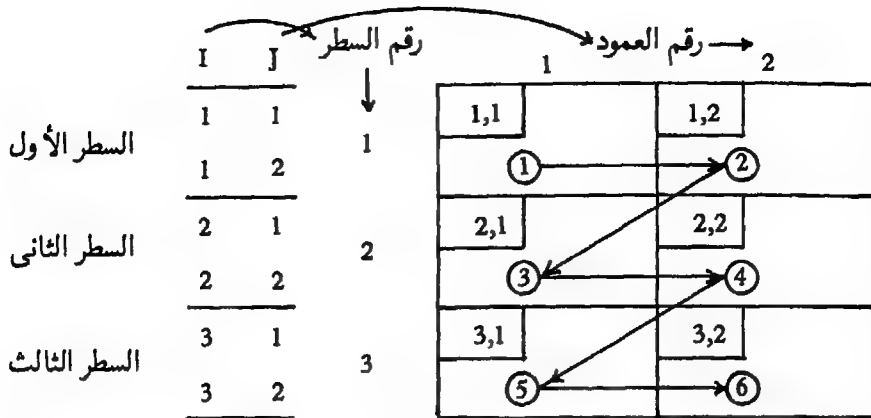
إذا أضفنا تعليمات الطباعة PRINT لمعرفة قيمة I، J أثناء تنفيذ الدائرة كالتالي :

```
100 FOR I = 1 TO 3
110 PRINT « 110 رقم السطر » ; I ; « J = » = J
150 FOR J = 1 TO 2
160 PRINT « 160 رقم السطر » ; « I = » , I ; « J = » ; I
200 NEXT J
210 PRINT « 210 رقم السطر » ; « I = » = I ; « J = » ; J
250 NEXT I
260 PRINT « 260 رقم السطر » ; « I = » ; I ; « J = » ; J
```

ولكان الناتج بعد تنفيذ البرنامج السابق :

ملاحظات	الناتج
بداية الدوارة المركبة — الدوارة الخارجية	$I = 1 \quad J = 0$ رقم السطر 110
بداية الدوارة الداخلية للمرة الأولى	$I = 1 \quad J = 1$ رقم السطر 160
	$I = 1 \quad J = 2$ رقم السطر 160
انتهاء دورة كاملة للدوارة الداخلية	$I = 1 \quad J = 3$ رقم السطر 210
بداية الدورة الثانية للدوارة الخارجية	$I = 2 \quad J = 3$ رقم السطر 110
بداية الدوارة الداخلية للمرة الثانية	$I = 2 \quad J = 1$ رقم السطر 160
	$I = 2 \quad J = 2$ رقم السطر 160
انتهاء الدورة الكاملة للدوارة الداخلية	$I = 2 \quad J = 3$ رقم السطر 210
بداية الدورة الثالثة للدوارة الخارجية	$I = 3 \quad J = 3$ رقم السطر 110
بداية الدوارة الداخلية للمرة الثالثة	$I = 3 \quad J = 1$ رقم السطر 160
	$I = 3 \quad J = 2$ رقم السطر 160
انتهاء الدورة الثالثة للدوارة الخارجية	$I = 3 \quad J = 3$ رقم السطر 210
انتهاء الدوارة المركبة	$I = 4 \quad J = 3$ رقم السطر 260

علاقة أرقام عدادات الدوارة بالمصفوفة ذات البعدين : لودقنا النظر إلى السطر ١٦٠ ، الواقع داخل الدوارة الداخلية ، وتبعنا تغير قيم I و J وقارناه بأرقام الصناديق في المصفوفة لوجدناه كالتالي :

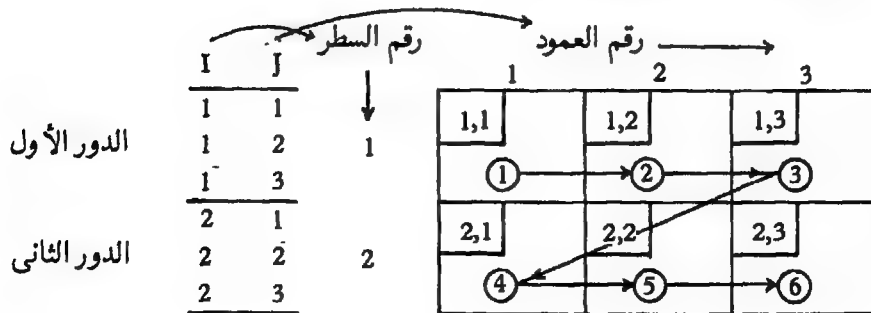


أى أن الدائرة المركبة بوضعها الحالى يمكن أن تستخدم لتمثيل مصفوفة ذات ثلاثة أسطر وعمودين ، و يتم ذلك عن طريق وضع الفاصلة (,) بين العددين I , J وإضافة اسم المصفوفة ، «S» مثلا ، فتكون :

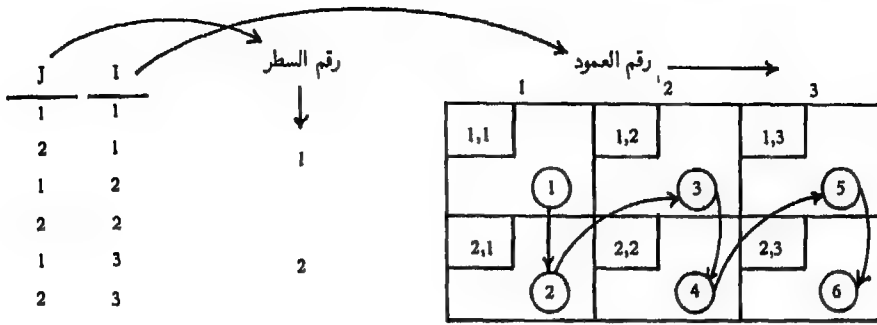
S (I , J) . ولاحظ تسلسل الإشارة إلى الصناديق :

① ← ② ← ③ ← ④ ← ⑤ ← ⑥ ، وذلك وفق تسلسل القيم المزدوجة I و J .

لاحظ أن رقم السطر في المصفوفة يمثل عدد الدائرة الخارجية ورقم العمود يمثل عدد الدائرة الداخلية ، وبناء على ذلك ، فإن التعامل مع مصفوفة ذات سطرين وثلاثة أعمدة ، أى على العكس من المصفوفة السابقة ، يتطلب جعل القيمة النهائية لعدد الدائرة الخارجية تساوى ٢ وقيمة عدد الدائرة الداخلية تساوى ٣ . وفي هذه الحالة تكون الإشارة إلى الصناديق كالتالى :



وهناك حل آخر وهو إبقاء قيمتى العدادين كما هما وإبدال مكان I مكان J بين القوسين، بحيث تشير I إلى رقم العمود و J إلى رقم السطر.
 $S(I, J)$ ، كما هو مبين في الشكل التالى :



لاحظ تسلسل الإشارة إلى الصناديق



وفي كل الأحوال، فإن المحافظة على تسلسل الإشارة إلى الصناديق بشكل أفقى تؤدي إلى وضوح أكثر وإلى أسلوب أبسر، لتخزين ومعالجة البيانات الخاصة بالمصفوفات ذات البعدين .

فملاحظة هذا التسلسل إذن أمر مهم جداً عند البدء في عمليات تخزين البيانات في الصناديق ومعالجتها .

وكما كان الشأن مع المصفوفة ذات البعد الواحد، فإن تعليمات بيسك قد تظهر قبل اسم المصفوفة، وذلك كما يلي :

```

READ      S (I,J)
INPUT     S (I,J)
PRINT     S (I,J)
LET       T = T + S (I,J)

```

وسنتطرق الآن إلى كيفية تطوير برنامج باستخدام تعليمات الدوارة المركبة في العمليات المصاحبة لاستخدام المصفوفات ذات البعدين .

تطوير برنامج باستخدام تعليمات الدوارة المركبة مع المصفوفات ذات البعدين :

سنعرض الآن لبرنامج مبسط لتوضيح كيفية التعامل مع المصفوفة ذات البعدين

مثال : (١٢ - ١) :

تؤجر المدينة الرياضية ثلاث صالات ، منها واحدة رئيسية واثنان ذواتا حجم أصغر ، فرعيتان . وهناك ناديان في المدينة : نادى الشباب ، ونادى الوحدة .

الهدف : معرفة مجموع الساعات لكل نادٍ ولكل صالة ، والمجموع الكلى لعدد الساعات ، وذلك كل شهر .

أولاً - خطوات الحل :

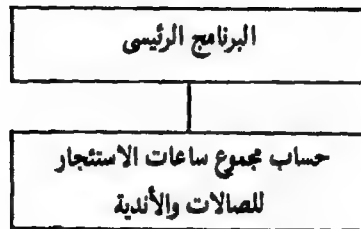
١ - الحصول على ساعات الإيجار لكل نادٍ في كل صالة ، وتخزينها في مصفوفة : عدد الأسطر ٢ يساوى عدد النوادي الرياضية ، وعدد الأعمدة ٣ يساوى عدد الصالات .

٢ - حساب مجموع الأعمدة والأسطر .

٣ - حساب المجموع الكلى لعدد الساعات .

٤ - طباعة النتائج .

ثانياً - الهيكل الهرمى :



وتوضح هذه البيانات في مصفوفة مبينة فيها أرقام الصناديق (ويطلق عليها خلايا Cells) كالتالى :

النواى		الأعمدة			
		الصالات			
الأسطر	I	J	1	2	3
			1,1	1,2	1,3
↓	1		20	40	25
	2		15	30	45

ويمكن قراءة هذه المصفوفة وتخزينها تحت اسم HR كما فى الجزء التالى :

```

FOR      I = 1 TO 2
FOR      J = 1 TO 3
READ     HR (I,J)
NEXT     J
NEXT     I
DATA     20, 40, 25
DATA     15, 30, 45

```

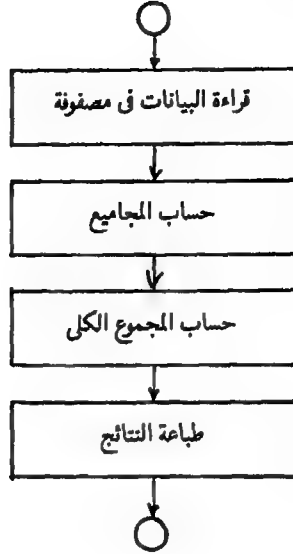
لاحظ التسلسل الأفقى لظهور البيانات فى عبارتى DATA ، وذلك وفق استخدام عددى الدوارة المركبة . ولوغيرنا سطر القراءة ليصبح كالتالى :

READ HR (J,I)

لتمت قراءة البيانات فى مصفوفة ذات ثلاثة أسطر وعمودين كما هو موضح فيما يلى :

1,1	20	1,2	15
2,1	40	2,2	30
3,1	25	3,2	45

ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرمجة :

ونورد هنا الشرح التفصيلي لتنفيذ هذه الأجزاء وما يصاحبها من عبارات البرمجة :

١ - الحصول على البيانات : كانت ساعات الاستئجار من قبل الأندية للصالات الثلاث لشهر رجب ١٤٠٧ هـ كما يلي :

النادي	الصالات		
	الصالات الرئيسية	الفرعية ١	الفرعية ٢
نادي الشباب	20	40	25
نادي الوحدة	15	30	45

أما إذا تم تغيير السطرين الخاصين بـ For ليصبحا كالتالي :

FOR I = 1 TO 3

FOR J = 1 TO 2

فإن ترتيب البيانات في عبارات DATA سيبقى كما هو في الوضع السابق، إلا أن المصفوفة سيكون لها ثلاثة أسطر ممثلة للصالات، وعمودان ممثلان للأندية، أى أن المصفوفة رقت بعكس الوضع السابق، كما هو موضح في الشكل التالى :

I	الأندية		J	الصالات
	1	2		
1	1,1	20	1,2	15
2	2,1	40	2,2	30
3	3,1	25	3,2	45

وكما تم ذكره سابقاً في هذا الفصل فإن الوضع الأول هو الأوضح والأسهل لتخزين البيانات في المصفوفات ومعالجتها. وهو ما سيتم اتباعه في هذا الفصل.

٢- الحصول على المجاميع : هناك أكثر من طريقة للحصول على المجاميع ومن هذه الطرق :

(أ) تخزين المجاميع في متغيرات مستقلة عن المصفوفة.

(ب) تخزين المجاميع في المصفوفة نفسها أى في السطر الثالث والعمود الرابع.

وسنبحث الأسلوب الأول هنا، وسنتعرض للأسلوب الثانى بعد تطوير البرنامج لهذا المثال بشكل كامل.

فلو فرضنا أن المتغيرات R1, R2, C1, C2, C3 ستحتوى على مجاميع عدد الساعات للأندية والصالات على التوالى، فستكون الحسابات كالتالى :

$$\begin{aligned}
 \text{للأسطر} \quad R1 &= HR(1,1) + HR(1,2) + HR(1,3) \\
 \text{الأندية} \quad R2 &= HR(2,1) + HR(2,2) + HR(2,3) \\
 \\
 \text{للأعمدة} \quad C1 &= HR(1,1) + HR(2,1) \\
 \text{الصالات} \quad C2 &= HR(1,2) + HR(2,2) \\
 \quad C3 &= HR(1,3) + HR(2,3)
 \end{aligned}$$

ولكن هذا الأسلوب سيبدو عديم الجدوى إذا ما افترضنا التعامل مع مصفوفة ذات بعدين كبيرين، ١٠٠ سطرو ٥٠ عموداً مثلاً. ومن الممكن اتباع ماتم شرحه عن العلاقة بين أرقام الصناديق، فتصبح المعادلات مع عبارات البرمجة كالتالى :

```

10 REM حساب مجموع الـ سطر
20 FOR J =1 TO 3
30 R1 = R1 + HR(1,J)
40 R2 = R2 + HR(2,J)
50 NEXT J
60 REM حساب مجموع الـ عمده
70 FOR I =1 TO 3
80 C1 = C1 + HR(I,1)
90 C2 = C2 + HR(I,2)
100 C3 = C3 + HR(I,3)
110 NEXT I

```

لاحظنا ثبات أرقام الأسطر وتغير أرقام الأعمدة فى الجزء الأول الخاص بحساب مجاميع الأسطر، وعلى العكس من ذلك فى الجزء الخاص بحساب مجاميع الأعمدة. ومن الممكن تتبع تنفيذ هذا الجزء كالتالى، ونبدأ بحساب مجاميع الأسطر:

الدورة	1	J	قيمة HR(I,J)	الأسطر R1	R2
1	1	1	20	20	
	2	1	15		15
2	1	2	40	60	
	2	2	30		45
3	1	3	25	85	
	2	3	45		90
المجموع النهائى				85	90

حيث يتم في كل دورة إضافة قيمتين من قيم المصفوفة HR إلى متغيرات المجاميع ذات العلاقة بكل سطر.

أما مجاميع الأعمدة فيمكن تتبعها كالتالي :

الدورة	I	J	HR (I,J)	C1	الأعمدة C2	C3
1	1	1	20	20		
	1	2	40		40	
	1	3	25			25
2	2	1	15	35		
	2	2	30		70	
	3	3	45			70
المجموع				35	70	70

حيث يتم إضافة ثلاث قيم من المصفوفة في كل دورة إلى قيمة المجاميع ذات العلاقة بكل عمود.

لاحظ اتجاه الأسهم بين قيم HR (I,J) والقيم المتجمعة في الأسطر والأعمدة والإشارة الحسابية من + أو =. ففي الشكل السابق خزنت القيمة 20 في C1 في الدورة الأولى، ثم أضيف لها 15 في الدورة الثانية، وهكذا مع بقية قيم المتغيرات الخاصة بمجاميع الأسطر والأعمدة.

٣- حساب المجموع الكلي : يتم تخزين كل قيمة من قيم الصناديق في متغير جامع ، كما هو موضح في الجزء التالي من البرنامج :

```
FOR      I = 1 TO 2  
  
FOR      J = 1 TO 3  
  
T      =  T  +  HR (I,J)  
  
NEXT    J  
  
NEXT    I
```

٤- كتابة النتائج : إن طباعة النتائج تحتاج إلى قليل من العناية . وسنكتفى هنا بطباعة أرقام الساعات كما في الجدول ، يليها أرقام المجاميع . وسنقدم فيما بعد برنامجاً لطباعة العناوين الموضحة لأرقام المصفوفة .

ويمكن تجميع الأجزاء السابقة في برنامج كامل كما في الشكل التالي :

شكل (١٢-١)

برنامج لحساب عدد الساعات المستأجرة من قبل الأندية لصالات المدينة الرياضية

```

10 REM البرنامج الرئيسي
20 GOSUB 90
30 END
90 REM البرنامج الفرعي
100 REM قراءة البيانات في مصفوفة HR
110 FOR I = 1 TO 2
120 FOR J = 1 TO 3
130 READ HR(I,J)
140 NEXT J
150 NEXT I
160 DATA 20,40,25
170 DATA 15,30,45
180 REM حساب مجاميع الأسطر
190 FOR J = 1 TO 2
200 R1 = R1 + HR(1,J):REM مجموع الأسطر الأول
210 R2 = R2 + HR(2,J):REM مجموع الأسطر الثاني
220 NEXT J
230 REM حساب مجاميع الأعمدة
235 FOR I = 1 TO 3
240 C1 = C1 + HR(I,1):REM مجموع الأعمدة الأول
250 C2 = C2 + HR(I,2):REM مجموع الأعمدة الثاني
260 C3 = C3 + HR(I,3):REM مجموع الأعمدة الثالث
270 NEXT I
280 REM حساب المجموع الكلي للساعات
290 FOR I = 1 TO 2
300 FOR J = 1 TO 3
310 T = T + HR(I,J)
320 NEXT J
330 NEXT I
340 REM طبع النتائج والبيانات المدخلة
350 FOR I = 1 TO 2
360 FOR J = 1 TO 3
370 PRINT USING "## ";HR(I,J);
380 NEXT J
390 PRINT " ":REM للانتقال للسطر الجديد
400 NEXT I
410 PRINT R1,"مجموع ساعات نادي الشباب"
420 PRINT R2,"مجموع ساعات نادي الوحدة"
430 PRINT C1,"مجموع ساعات الصالة الكبرى"
440 PRINT C2,"مجموع ساعات الصالة الفرعية الأولى"
450 PRINT C3,"مجموع ساعات الصالة الفرعية الثانية"
460 PRINT T,"المجموع الكلي للساعات المستأجرة"
470 RETURN

```


وفي حالة تنفيذ البرنامج سنحصل على النتائج التالية .

20	40	25
15	30	45
60	مجموع ساعات نادي الشباب	
45	مجموع ساعات نادي الوحدة	
35	مجموع ساعات الصالة الكبرى	
70	مجموع ساعات الصالة الفرعية الاولى	
70	مجموع ساعات الصالة الفرعية الثانية	
175	المجموع الكلي للساعات المستأجرة	

ملاحظات على البرنامج :

١ - إن الأجزاء الثلاثة الأولى من البرنامج الفرعى يمكن أن تجمع حقاً في دوائر مركبة واحدة. وتوضيح ذلك أنه عند قراءة كل قيمة من قيم ساعات الاستئجار يتم تخزينها في الخلية المناسبة، وإضافتها إلى مجموع السطر والعمود ذوى العلاقة وإلى القيمة الكلية للساعات.

ويمكن تعديل البرنامج السابق ليتوافق مع الأسلوب الجديد بإزالة الأسطر

التالية :

300, 290, 270, 235, 220, 190, 150, 140

٢ - إذا أردنا تخزين مجموع الأسطر في العمود الأخير ومجموع الأعمدة في السطر الأخير، يصبح شكل المصفوفة النهائية بعد إجراء عمليات الجمع كالتالى :

النادى	الصالات			
	الرئيسية	الفرعية ١	الفرعية ٢	المجموع
الشباب	1,1 20	1,2 40	1,3 25	1,4 85
الوحدة	2,1 15	2,2 30	2,3 45	2,4 90
المجموع	3,1 35	3,2 70	3,3 70	3,4 175

وباستخدام العلاقة بين تغيرات أرقام الصناديق، يمكن تطوير العبارات الخاصة بتجميع الأسطر والأعمدة كالتالى :

```
REM          حساب مجموع الأسطر والأعمدة
HR (1,4) = HR (1,4) + HR (1,J)
HR (3,J) = HR (3,J) + HR (1,J)
REM          حساب المجموع الكلى لساعات الاستئجار
HR (3,4) = HR (3,4) + HR (1,J)
```

ولإحلال هذه الأسطر فى البرنامج مكان الأسطر الأخرى ، احذف الأسطر من 200 - 260 ، ومن ثم أضف الأسطر الجديدة .

٣- إذا أريد طباعة الجداول كما فى شكلها السابق ، يجب تخزين أسماء الأندية مع كلمة «المجموع» فى مصفوفة مستقلة ذات بعد واحد ، ذلك لأن قوانين لغة البيسك تمنع الجمع بين بيانات حسابية وغير حسابية (حرفية) فى نفس المصفوفة . ويمكن تحقيق ذلك كما فى الجزء التالى :

```
REM          قراءة أسماء الأندية كلمة المجموع
FOR K = 1 TO 3
READ N$(K)
NEXT K
DATA «المجموع» و «الوحدة» و «الشباب»
```

وهذا التخزين لأسماء الأندية ضرورى ، لنتمكن من طباعة اسم النادى قبل طباعة الساعات الخاصة به . أما طباعة أسماء الصالات الرياضية فأمر سهل ، إذ يمكن طباعتها فى سطر واحد قبل بدء طبع أسماء الأندية والصالات .

ولطباعة الجدول، من الأفضل أن تحدد أماكن طبع العناوين والأرقام على الشاشة، كما في الشكل التالي :

النادى	الصالات			
	الرئيسية	الفرعية ١	الفرعية ٢	المجموع
الشباب	xx	xx	xx	xxx
الوحدة	xx	xx	xx	xxx
المجموع	xx	xx	xx	xxx

وسنبداً بكتابة العناوين ثم طباعة الجدول .

التعليمات التالية تطبع السطر الأول والثاني من العناوين :

PRINT TAB (33); «الصالات»

PRINT TAB (10); «النادى» TAB (22) ; «الرئيسية» ; TAB (32);

«المجموع» ; TAB (52); «الفرعية ٢» ; TAB (42) ; «الفرعية ١» ;

والتعليمات التالية تقوم بطباعة الجدول مع أسماء النوادي :

FOR I = 1 TO 3

PRINT TAB (12); N\$(I) ;

FOR J = 1 TO 4

PRINT « » ; HR (I,J);

NEXT J

PRINT

NEXT I

ولتعديل البرنامج في الشكل (١٢-١)، احذف الأسطر من 340 إلى 460 وأضف الأسطر البديلة السابقة، لاحظ أن البرنامج في هذا الشكل قد عدل أكثر من مرة إلى الآن، وسنورده هنا في شكله النهائي :

شكل (١٢ - ١)

برنامج لقراءة البيانات في مصفوفة ذات بعدين وحساب
مجاميع الأسطر والأعمدة وطباعة المصفوفة مع النتائج

```

10 REM البرنامج الرئيسي
20 GOSUB 40
30 END
40 REM البرنامج الفرعي
50 REM      HR مصفوفة البيانات في مصفوفة
60 FOR I = 1 TO 2
70 FOR J = 1 TO 3
80 READ HR(I,J)
90 NEXT J
100 NEXT I
110 DATA 20,40,25
120 DATA 15,30,45
130 REM قراءة أسماء الأندية وكلمة المجموع
140 FOR K = 1 TO 3
150 READ N*(K)
160 NEXT K
170 DATA "المجموع", "الوحدة", "الشباب"
180 FOR I = 1 TO 2
190 FOR J = 1 TO 3
200 REM حساب مجاميع الأسطر والأعمدة
210 HR(I,4) = HR(I,4) + HR(I,J)
220 HR(3,J) = HR(3,J) + HR(I,J)
230 REM حساب المجموع الكلي للمساحات المستطيلة
240 HR(3,4) = HR(3,4) + HR(I,J)
250 NEXT J
260 NEXT I
270 REM طباعة السطر الأول والعناوين
280 PRINT TAB(33); "الصلات"
290 PRINT TAB(12); "النادي"; TAB(25); "الرئيسية"; TAB(37); "الفرعية ١"; TAB(51); "الفرعية ٢"
300 PRINT "المجموع"; TAB(65); "الفرعية ٢"
310 REM طباعة الجدول مع أسماء الأندية
320 FOR I = 1 TO 3
330 PRINT TAB(12); N*(I);
340 FOR J = 1 TO 4
350 PRINT "      "; HR(I,J);
360 NEXT J
370 PRINT " "
380 NEXT I
390 RETURN

```

وعند تنفيذ البرنامج سنحصل على النتائج التالية :

المجموع	الصالات			النادي
	الفرعية ٢	الفرعية ١	الرئيسية	
85	25	40	20	الشباب
90	45	30	15	الوحدة
175	70	70	35	المجموع

وسنضيف الآن هدفاً آخر للمثال السابق يدعو إلى حساب ما سيدفعه كل نادٍ مقابل ساعات الاستئجار، إذا علم أن كل نادٍ سيدفع مبلغاً محدداً لكل ساعة، كما هو مبين في الجدول التالي :

إجمالي ساعات الاستئجار في الصالات الرياضية بالريال السعودي .

النادي	الصالات		
	الفرعية ٢	الفرعية ١	الرئيسية
الشباب	190	230	400
الوحدة	200	220	320

حيث يقرر هذا المبلغ مركز النادي في الدوري الممتاز، وقد فاز نادى الشباب بالدورى لهذه السنة، وبناء على ذلك فأجر الساعة له أعلى من نادى الوحدة الذى حقق المركز الرابع، إلا فى الصالة الفرعية ٢ .

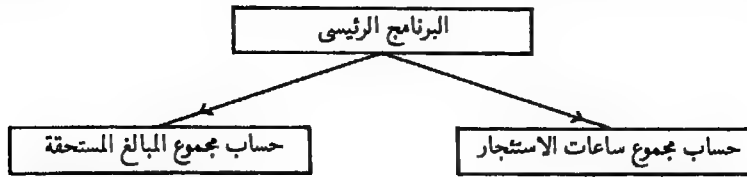
ولتحقيق هذا الهدف، نضيف الخطوات التالية :

٥ — قراءة مصفوفة الأجر للساعة الواحدة من ساعات الاستئجار.

٦ — حساب مجموع ما سيدفعه كل نادٍ والدخل لكل صالة .

٧ — الطباعة .

وسنعمل للبرنامج السابق جزأين رئيسيين كما هو مبين في الهيكل الهرمي التالي :



وقد تم تنفيذ الجزء الأول وسنتم الآن الجزء الثاني وفق الخطوات السابقة وباستخدام نفس نمط البرمجة التركيبية للجزء الأول.

٥ - قراءة المصفوفة :

800 FOR I = 1 TO 2

810 FOR J = 1 TO 3

820 READ RT (I,J)

830 NEXT J

840 NEXT I

850 DATA 400, 230, 190

860 DATA 320, 220, 200

895 REM

٦ - حساب المجموع المستحق :

900 FOR I = 1 TO 2

910 FOR J = 1 TO 3

920 PY (I,J) = HR (I,J) * RT (I,J)

930 PY (I,4) = PY (I,4) + PY (I,J)

940 PY (3,J) = PY (3,J) + PY (I,J)

950 TT = TT + PY (I,J)

960 NEXT J

970 NEXT I

٧ - الطباعة : 1000 REM

1010 PRINT TAB (40) ; «مجموع مستحقات النوادي والإيرادات»

1020 PRINT TAB (10); «النوادي»

TAB (52);

1050 FOR I = 1 TO 3

1060 PRINT TAB (12); N\$(I);

1070 FOR J = 1 TO 4

1080 PRINT « »; PY (I,J) ;

1090 NEXT J

1100 PRINT

1110 NEXT I

1200 RETURN

ولتنفيذ هذا الجزء ، أضف الأسطر السابقة إلى البرنامج في الشكل (١٢ - ١١) ،

مع العبارة التالية : 25 GOSUB 800

وفي حالة تنفيذ البرنامج المعدل سنحصل على النتائج التالية :

الصالات

المجموع	الفرعية ٢	الفرعية ١	الرئيسية	النادي
85	25	40	20	الشباب
90	45	30	15	الوحدة
175	70	70	35	المجموع

مجموع مستحقات النوادي وإيرادات الصالات

الصالات

المجموع	الفرعية ٢	الفرعية ١	الرئيسية	النوادي
85	25	40	20	الشباب
90	45	30	15	الوحدة
175	70	70	35	المجموع

إجراء العمليات الرياضية على المصفوفات واستخدامات تعليمة مصفوفة MAT :

تعرضنا في الجزء السابق من هذا الفصل لكيفية معالجة البيانات المخزنة في مصفوفة ذات بعدين عن طريق الدوارة المركبة، لكن التطبيقات العلمية والرياضية قد تتطلب معالجة معقدة على المصفوفات، بحيث تجعل الاعتماد على تعليمات الدوارة المركبة قليل الجدوى.

وستتعرض في هذا الجزء من الفصل لاستخدام تعليمة MAT، والتي تسهل المعالجة الخاصة بالمصفوفات. وتنقسم هذه الاستعمالات إلى ثلاثة أقسام رئيسية :

- ١ — استخدامات في الإدخال والإخراج.
 - ٢ — استخدامات في العمليات الحسابية من إضافة أو ضرب مصفوفة في رقم واحد محدد، أو جمع وضرب وطرح مصفوفتين.
 - ٣ — استخدامات في عمليات خاصة مثل إيجاد منقول ومعكوس المصفوفة.
- وسيتتم شرح جميع هذه الاستخدامات مع التركيز على الاستخدام الأول والثاني لمناسبتهما لموضوع الكتاب.

تطوير برنامج باستخدام تعليمة MAT :

مثال (١٢ - ٢) :

الهدف : دائرة الأرصاد الجوية في مدينة معينة ترغب في إيجاد متوسط درجات الحرارة لكل أسبوع من الأسابيع الأربعة ومن ثم المتوسط العام للأسابيع الأربعة.

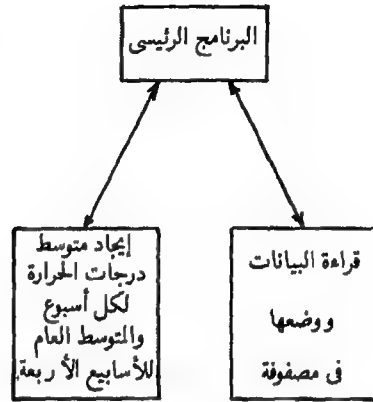
أولاً — خطوات الحل :

- ١ — الحصول على درجات الحرارة للأسبوع.
- ٢ — إيجاد متوسط درجات الحرارة للأسبوع.

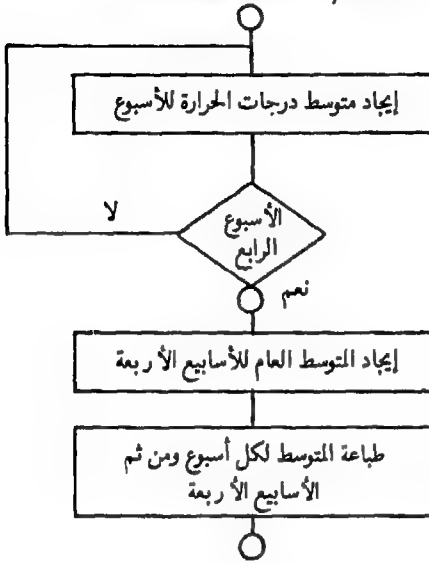
٣- تكرار الخطوتين ١ و ٢ للأسابيع الأربعة.

٤- إيجاد متوسط درجات الحرارة للأسابيع الأربعة.

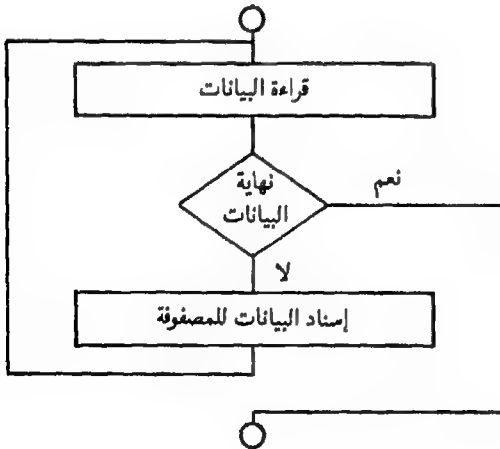
ثانياً- الهيكل الهرمي :



ثالثاً- نمط البرمجة الهيكلية :



قراءة درجات الحرارة ووضعها في مصفوفة



رابعاً - سيتم تطوير البرنامج أولاً باستخدام تعليمات الدوارة المركبة ، كما في الشكل التالي ، ومن ثم سيعديل البرنامج باستخدام تعليمة مصفوفة MAT ، وذلك من أجل المقارنة بين الأسلوبين .

شكل (١٢ - ٢)

```

برنامج لإيجاد متوسط درجات الحرارة لكل أسبوع ومن ثم الأسابيع الأربعة

10 REM البرنامج الرئيسي
20 DIM T(4,7),SW(4)
30 GOSUB 60
40 GOSUB 120
50 END
60 REM T لقراءة درجات الحرارة ووضعها في مصفوفة
70 FOR W = 1 TO 4:REM عدد الأسابيع
80 FOR D = 1 TO 7:REM عدد الأيام
90 READ T(W,D):REM لقراءة درجة الحرارة لليوم
100 NEXT D : NEXT W
110 RETURN
120 REM برنامج فرعي لا يجاد متوسط درجات الحرارة لكل اسبوع
130 FOR W = 1 TO 4 : FOR D= 1 TO 7: S = S + T (W,D)
140 SW(W) = SW(W) + T(W,D): NEXT D : NEXT W
150 FOR W = 1 TO 4
160 LET A = SW(W) / 7:REM لا يجاد المتوسط لكل اسبوع
170 PRINT USING "##.## = % "متوسط درجات الحرارة لا سبوع رقم A,W
180 NEXT W
190 REM لطباعة المتوسط العام لدرجات الحرارة
200 PRINT "
210 PRINT "-----"
220 DATA 29,30,31,32,28,27,28,25,25,27,25,26,27,28
230 DATA 30,31,31,30,32,31,30,33,32,32,33,33,31,30
240 RETURN

```

وفي حالة تنفيذ البرنامج سنحصل على النتائج التالية :

29.29	=	1	متوسط درجات الحرارة لا سبوع رقم
26.14	=	2	متوسط درجات الحرارة لا سبوع رقم
30.71	=	3	متوسط درجات الحرارة لا سبوع رقم
32.00	=	4	متوسط درجات الحرارة لا سبوع رقم

29.53572 = المتوسط العام لدرجات الحرارة في الأسابيع الأربعة

وكما مر ذكره في الجزء السابق من هذا الفصل ، فإن قراءة البيانات ووضعها في مصفوفة أو إخراج البيانات من مصفوفة ، يتم بالتعامل مع كل خلية (عمود وسطر) من خلايا السطر الأول ، وبعد الانتهاء منها يتم التعامل مع خلايا السطر الثانى... وهكذا... ويتم ذلك باستخدام تعليمة For - Next حيث يحدد بها عدد الأسطر والأعمدة (أبعاد المصفوفة) التى تتكون منها المصفوفة .

وباستخدام تعليمة MAT - وهى اختصار لكلمة مصفوفة MATRIX - نستغنى إلى حد كبير عن استخدام تعليمة For - Next فى التعامل مع المصفوفة .

والشكل العام لاستخدام تعليمة MAT لقراءة وطباعة المصفوفة هو كالتالى :

الشكل العام لعبارة تعليمة مصفوفة MAT عند استخدامها فى القراءة والكتابة			
xxx	MAT	xxxx	x
↓	↓	↓	↓
رقم السطر	تعليمة مصفوفة	إحدى تعليمات الإدخال أو الإخراج	* اسم المصفوفة

* ملاحظة :

اسم المصفوفة هو الاسم الذى يلى تعليمة DIM والذى تحدد بعده (بين قوسين) أبعاد المصفوفة ، فمثلاً فى مثالنا السابق شكل (١٢ - ٢) فى العبارة رقم (10) والتى حددت بها أبعاد وأسماء كل من المصفوفة T وأبعادها أربعة أسطر وسبعة أعمدة والمصفوفة SW وأبعادها أربعة سطور وعمود واحد . ونستخدم هذين الاسمين T و SW للتعامل مع هاتين المصفوفتين فى حالة استخدام تعليمة MAT لقراءة درجات الحرارة للأسابيع الأربعة نستخدم العبارة التالية :

60 MAT READ T

لتحل مكان كل من العبارات 60 إلى 90 .

كما يمكننا طباعة درجات الحرارة للأسابيع الأربعة باستخدام تعليمة MAT كما

هو مبين بالعبارة التالية : 20 MAT PRINT T

والشكل العام حسب هذا الاستعمال هو كالتالى :

الشكل العام لعبارة MAT للاستخدامات الحسابية					
xxx	MAT	X1	=	X2	*
↓	↓	↓		↓	↓
↓	↓	↓		↓	↓
رقم السطر	تعليمة مصفوفة	المصفوفة الجديدة		مصفوفة حالية	رمز العملية الحسابية
					+
					-
					↓
					مصفوفة أو رقم

فإذا أردنا تحويل درجات الحرارة من المئوية إلى الفهرنهايتية نعمل على ضرب كل خلية من خلايا المصفوفة فى القيمة $\frac{5}{9}$ ومن ثم نضيف القيمة ٣٢ إلى ناتج الضرب .

مقارنة بين العبارات اللازمة لتنفيذ عملية التحويل :

ب) باستخدام تعليمة MAT	أ) باستخدام تعليمة FOR... NEXT
10 DIM C(4,7), F(4,7), T(4,7)	10 DIM C(4,7), F(4,7)
40 MAT READ C	20 FOR I = 1 TO 4
45 LET D = 5/9	30 FOR J = 1 TO 7
50 MAT T = C * D	40 READ C (I,J)
55 MAT F = T + 32	50 LET F (I,J) = C(I,J) * (5/9) + 32
60 MAT PRINT F	60 PRINT F (I,J)
	70 NEXT J
	80 PRINT
	90 NEXT I

لاحظ الفروقات ما بين كل من الطريقتين (أ و ب).

١ - عدد العبارات التي احتجنا إليها باستخدام تعليمة FOR... NEXT أكثر، وهذه إحدى الفوائد من استخدام تعليمة MAT حيث تعمل على تقليل عدد العبارات اللازمة لتنفيذ بعض العمليات .

٢ - تم التحويل من مئوية إلى فهرنهايتية بعبارة واحدة عندما استخدمنا تعليمة FOR...NEXT (عبارة 50)، وفي حالة استخدام تعليمة MAT احتجنا إلى عبارة لنسند بها ناتج قيمة $\frac{5}{9}$ للمتغير D (عبارة 30)، ومن ثم تمت عملية ضرب كل خلية من خلايا المصفوفة C بقيمة المتغير D وإسناد القيم إلى المصفوفة T. وبعد الإسناد أضيفت القيمة ٣٢ لكل خلية من خلايا المصفوفة T. وأسندت جميع خلاياها للمصفوفة F.

والسبب في ذلك يعود إلى عدم إمكانية إجراء أكثر من عملية حسابية في كل عبارة يتم استخدام تعليمة MAT بها، ولذا يتم التحايل على ذلك باستخدام مصفوفة مؤقتة، ويجب أن يكون حجم هذه المصفوفة (أى عدد السطور وعدد الأعمدة) مساوياً لحجم المصفوفة التى ستسند إليها القيمة النهائية.

٣ - إمكانية إجراء عملية ضرب قيمة في كل خلية من خلايا المصفوفة باستخدام تعليمة MAT (عبارة 40).

٤ - إمكانية إضافة (جمع) قيمة لكل خلية من خلايا المصفوفة باستخدام MAT (عبارة 50).

العمليات الرياضية التى يمكن إجراؤها على المصفوفات باستخدام تعليمة MAT :
أولاً - إسناد قيمة ثابتة لكل خلية من خلايا المصفوفة، فمثلاً إذا أردنا إسناد القيمة صفر (0) لكل خلايا المصفوفة يمكننا عمل ذلك باستخدام العبارتين التاليتين :

10 DIM A (2,3)

20 MAT A = 0

وبعد تنفيذ العبارتين ستصبح المصفوفة A كالتالي :

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

ثانياً - إسناد قيم خلايا مصفوفة إلى مصفوفة أخرى :

$$\text{فإذا أردنا إسناد القيم } \begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix} \text{ للمصفوفة A}$$

ومن ثم إسناد هذه القيم للمصفوفة B نستخدم العبارات التالية :

```
10 DIM A(2,2), B(2,2)
20 MAT INPUT A
30 MAT B = A
```

وبعد تنفيذ هذه العبارات سيطلب منا إدخال قيم المصفوفة A وفي حالة إدخال

$$B = \begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix} \text{ القيم 2,4,6,8 فإن ناتج التنفيذ سيصبح}$$

ثالثاً - جمع قيم خلايا مصفوفتين، فإذا كانت لدينا المصفوفتان أ، ب

فإن عملية الجمع تتم بإضافة قيمة الخانة الأولى (سطر ١ وعمود ١) من المصفوفة الأولى إلى قيمة الخانة الأولى (سطر ١ وعمود ١) من المصفوفة الثانية وإسناد الناتج للخانة الأولى من المصفوفة الثالثة (الجديدة)، ثم يتم إضافة قيمة الخانة الثانية (سطر ١ وعمود ٢) من المصفوفة الأولى إلى قيمة الخانة الثانية من المصفوفة الثانية، وإسناد الناتج إلى قيمة الخانة الثانية من المصفوفة الثالثة، أي جمع الخلايا المتقابلة كما هو مبين

فيما يلي :

$$C = A + B$$

$$\begin{bmatrix} \text{الأولى} & \text{الثانية} \\ \text{الثالثة} & \text{الرابعة} \end{bmatrix} C = \begin{bmatrix} \text{الأولى} & \text{الثانية} \\ \text{الثالثة} & \text{الرابعة} \end{bmatrix} B + \begin{bmatrix} \text{الأولى} & \text{الثانية} \\ \text{الثالثة} & \text{الرابعة} \end{bmatrix} A$$

وإذا كانت قيمة كل من $A = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$ $B = \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix}$ فإن $C = \begin{bmatrix} 6 & 9 \\ 6 & 8 \end{bmatrix}$

و يتم ذلك باستخدام العبارات التالية :

10 DIM A (2,2), C (2,2)

20 MAT READ A

30 MAT READ B

40 MAT C = A + B

50 MAT PRINT A,B,C

60 DATA 2,3,4,5

70 DATA 4,6,2,3

80 END

وبعد تنفيذ هذه العبارات ستظهر لنا النتائج التالية :

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} + \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 6 & 9 \\ 6 & 8 \end{bmatrix}$$

$\begin{matrix} \text{A} & + & \text{B} & = & \text{C} \\ \text{قيم خلايا المصفوفة A} & & \text{قيم خلايا المصفوفة B} & & \text{قيم خلايا المصفوفة C} \\ & & & & \text{(ناتج عملية الجمع)} \end{matrix}$

Diagram illustrating the addition of matrices A and B to get C:

- Matrix A: $\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$
- Matrix B: $\begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix}$
- Matrix C: $\begin{bmatrix} 6 & 9 \\ 6 & 8 \end{bmatrix}$
- Arrows indicate the element-wise addition:
 - Top-left: $2+4=6$
 - Top-right: $3+6=9$
 - Bottom-left: $4+2=6$
 - Bottom-right: $5+3=8$

رابعاً – طرح قيم خلايا مصفوفة من أخرى وشروط عملية الطرح هي نفس شروط عملية الجمع أى :

١ – يجب أن تكون كلتا المصفوفتين متطابقتين فى الحجم (عدد الأسطر وعدد الأعمدة).

٢- يتم طرح قيمة كل خلية من المصفوفة الثانية من قيمة الخلية المقابلة لها في المصفوفة الأولى كما هو مبين فيما يلي :

$$C = A - B$$

$$C = \begin{bmatrix} 2-4 & 3-6 \\ 4-2 & 5-3 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix} \quad A = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

ويتم ذلك باستخدام العبارات التالية :

10 DIM A(2,2), C(2,2)

20 MAT READ A

30 MAT READ B

40 MAT C = A - B

50 MAT PRINT A,B,C

60 DATA 2,3,4,5

70 DATA 4,6,2,3

80 END

وبعد تنفيذ هذه العبارات ستظهر لنا النتائج التالية :

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \quad \begin{bmatrix} 4 & 6 \\ 2 & 3 \end{bmatrix} \quad \begin{bmatrix} -2 & -3 \\ 2 & 2 \end{bmatrix}$$

قيمة خلايا المصفوفة A قيمة خلايا المصفوفة B قيمة خلايا المصفوفة C
ناتج عملية الطرح

خامساً - ضرب قيم خلايا مصفوفتين :

إن شروط عملية الضرب التي يجب توافرها في المصفوفات تختلف عن الشروط في كل من عمليتي الجمع والطرح ، ويعود السبب إلى كيفية تنفيذ عملية الضرب وتحديد عدد خلايا المصفوفة الجديدة.

فالمصفوفة التى ستحتوى خلاياها على ناتج الضرب تحدد أبعادها حسب التالى :

١ — عدد أسطرها يكون مساوياً لعدد أسطر المصفوفة الأولى .

٢ — عدد أعمدتها يكون مساوياً لعدد أعمدة المصفوفة الثانية .

وقبل القيام بعملية الضرب يجب تحقيق الشرط الرئيسى ، وهو أن يكون عدد أعمدة المصفوفة الأولى مساوياً لعدد أسطر المصفوفة الثانية ، وإلا فلن تتحقق عملية الضرب .
فمثلاً — إذا كانت لدينا المصفوفتان أ و ب وأبعاد كل منهما كالتالى :

المصفوفة	عدد الأسطر	عدد الأعمدة
أ	٣	٢
ب	٣	٢

لاحظ هنا أن عدد أعمدة المصفوفة أ لم تساوِ عدد أسطر (ب) لهذا لن تتم عملية الضرب

أما إذا كانت أبعادها كالتالى :

أ	٣	٢
ب	٢	٣

فسوف تتم عملية الضرب لأن عدد أعمدة المصفوفة أ (٢) مساوٍ لعدد أسطر المصفوفة ب أى أن الشرط الرئيسى قد تحقق ، وعليه ستكون أبعاد المصفوفة الجديدة كالتالى :

$$\begin{aligned} \text{عدد الأسطر} &= \text{عدد أسطر المصفوفة أ} = ٣ \\ \text{عدد الأعمدة} &= \text{عدد أعمدة المصفوفة ب} = ٢ \end{aligned}$$

ملاحظة :

بإمكاننا ضرب المصفوفة أ فى المصفوفة ب لكن لا يمكننا ضرب المصفوفة ب فى المصفوفة أ لعدم تحقق الشرط الرئيسى .

فإذا كان لدينا المصفوفتان أ و ب وكانت قيمة كل منهما كالتالي :

$$\begin{bmatrix} \text{د} & \text{هـ} \\ \text{و} & \text{ع} \end{bmatrix} = \text{و ب} = \begin{bmatrix} \text{س} & \text{ص} \\ \text{ی} & \text{ل} \end{bmatrix} \text{فان ج} = \begin{bmatrix} \text{د} \times \text{س} + \text{هـ} \times \text{ی} & \text{د} \times \text{ص} + \text{هـ} \times \text{ل} \\ \text{و} \times \text{س} + \text{ع} \times \text{ی} & \text{و} \times \text{ص} + \text{ع} \times \text{ل} \end{bmatrix}$$

ويتم ذلك باستخدام العبارات التالية :

```
10  DIM A (2,2), B(2,2), C(2,2)
```

20 MAT READ A

30 MAT READ B

40. MAT C = A * B

```
50 MAT PRINT A,B,C
```

60 DATA 2,3,4,5

70 DATA 4,6,2,3

80 END

وبعد تنفيذ هذه العبارات ستظهر لنا النتائج التالية :

2	3
4	5

قيمة خلايا المصفوفة A

4	6
2	3

قيمة خلايا المصفوفة B

14	21
-	
26	39

قيمة خلايا المصفوفة C

'ناتج عملية الضرب

سادساً - إيجاد منقول المصفوفة **: TRANSPOSE OF THE MATRIX**

ومنقول المصفوفة يعنى أن قيم خلايا الأعمدة في مصفوفة تصبح قيم خلايا السطور المقابلة لها في مصفوفة أخرى، أى (قيم خلايا العمود الأول في المصفوفة الأولى تصبح

قيم خلايا السطر الأول في المصفوفة الثانية وهكذا...) ويتم ذلك باستخدام التعليمة TRN كما هو مبين بالشكل التالي :

الشكل العام لعبارة تعليمة منقول المصفوفة			
TRN			
XXX	MAT	B =	TRN (A)
↓	↓	↓	↓
رقم السطر	تعليمة مصفوفة	اسم المصفوفة التي ستنقل إلى أسطرها قيم أعمدة المصفوفة A	تعليمة منقول
			اسم المصفوفة التي ستنقل قيم أسطرها لتصبح قيم أعمدة المصفوفة B

وهنا يجب أن يكون عدد أسطر المصفوفة التي ستنقل قيم خلاياها مساوياً لعدد أعمدة المصفوفة التي سينقل إليها.

فإذا كانت لدينا المصفوفة (أ) وقيم خلاياها كالتالي :

$$\begin{bmatrix} 6 & 5 & 2 \\ 8 & 3 & 4 \end{bmatrix} \quad \text{فإن منقولها سيكون} \quad \begin{bmatrix} 4 & 2 \\ 3 & 5 \\ 8 & 6 \end{bmatrix} = A$$

و يتم ذلك باستخدام العبارات التالية :

```

10  DIM A (3,2), B(2,3)
20  MAT READ A
30  MAT B = TRN (A)
40  MAT PRINT A,B
50  DATA 2,4,5,3,6,8
60  END

```

وبعد تنفيذ هذه العبارات ستظهر لنا النتائج التالية :

$$\begin{bmatrix} 2 & 4 \\ 5 & 3 \\ 6 & 8 \end{bmatrix} \quad \begin{bmatrix} 2 & 5 & 6 \\ 4 & 3 & 8 \end{bmatrix}$$

قيمة خلايا المصفوفة B والتي تساوى منقول المصفوفة A قيمة خلايا المصفوفة A

سابعاً - إيجاد معكوس / نظير المصفوفة INVERSE OF THE MATRIX :

هناك العديد من الطرق والنظريات المستخدمة لإيجاد معكوس المصفوفة، ولكن جميع هذه النظريات تخرج بنتيجة واحدة وهي أن نظير المصفوفة يتمثل في القاعدة التالية :

$$A * B = B * A = I$$

حيث إن I هو نظير المصفوفة الأصلية A. أما المصفوفة B فيجب أن تكون مساوية للمصفوفة A من حيث عدد الأسطر وعدد الأعمدة، أى أن أبعادها متساوية. كما أنه يشترط في المصفوفة المراد إيجاد نظيرها أن تكون أبعادها متساوية، أى أن عدد الأعمدة يجب أن يكون مساوياً لعدد الأسطر.

ولإيجاد نظير المصفوفة نستخدم تعليمة نظير INV وهي اختصار لكلمة INVERSE كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة نظير		INV
اسم المصفوفة التى لدينا والمراد إيجاد نظيرها .	تعليمة نظير	INV (A)
=		
المصفوفة التى سيسند إليها نظير المصفوفة A	B	
رقم السطر	XXXX	

فإذا قمنا بتنفيذ هذه العبارة فستكون النتائج كالتالى :

المصفوفة الأصلية	
4	2
3	1
نظير المصفوفة	
-0.5	1
1.5	-2

تطوير برنامج

تجارى باستخدام تعليمات الدوارة المركبة وتعليمة MAT :

مثال (١٢ - ٣) :

شركة تنتج ثلاثة أصناف باستخدام ثلاث آلات ويتم تحديد نسبة الربح المطلوبة للقطعة حسب أسلوب إنتاجها .
الهدف : إيجاد سعر البيع للوحدة المنتجة :

أولاً - خطوات الحل :

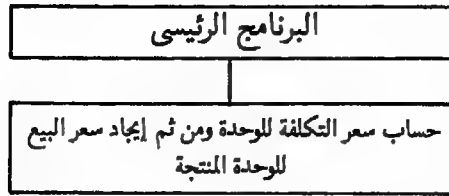
١ - قراءة مصفوفات التكلفة الكلية، عدد الوحدات المنتجة، نسبة الربح المطلوبة.

٢ - حساب سعر التكلفة للوحدة حيث إنه يساوى : $\frac{\text{التكلفة الكلية}}{\text{عدد الوحدات}}$

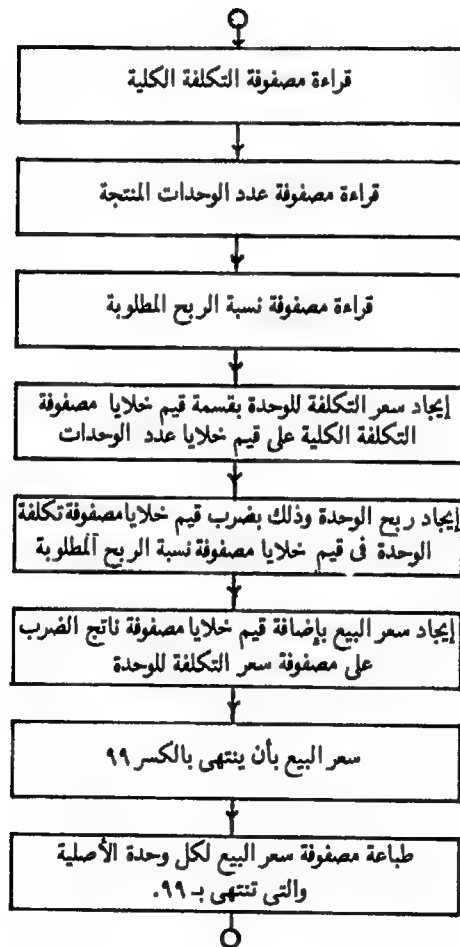
٣ - حساب سعر البيع = سعر التكلفة للوحدة + نسبة الربح المطلوبة بحيث تكون الأسعار ٩٩,٩٩.XX.

٤ - طباعة النتائج .

ثانياً - الهيكل الهرمي للبرنامج :



ثالثاً - نمط البرمجة التركيبية :



رابعاً: سيتم تنفيذ البرنامج على الحاسب الكبير IBM 3033 نظراً لعدم توفر تعليمة MAT على الحاسبات الشخصية (IBC / XT).

شكل (١٢ - ٣)

برنامج لإيجاد سعر البيع للوحدة من الأصناف الثلاثة المنتجة

```

10 REM البرنامج الرئيسي
20 REM استدعاء برنامج فرعي
30 GOSUB 50
40 END
50 REM برنامج فرعي لإيجاد سعر البيع للوحدة
60 DIM C(3,2),U(3,2),P(3,2),R(3,2),T(3,2),S(3,2),N(3,2)
70 REM قراءة و طباعة المصفوفات الثلاثة وهم
80 REM التكلفة الكليه , الوحدات المنتجة , نسبة الربح
90 MAT READ C,U,P
100 MAT PRINT C,U,P
110 REM لإيجاد التكلفة للوحدة
120 FOR I = 1 TO 3
130 FOR J = 1 TO 2
140 LET R(I,J) = C(I,J) / U(I,J)
150 LET T(I,J) = R(I,J) * P(I,J)
160 NEXT J
170 NEXT I
180 REM لطباعة مصفوفتي تكلفة الوحدة و الارباح
190 MAT PRINT R
200 MAT PRINT T
210 REM لإيجاد سعر البيع باضافة خلايا ناتج الضرب على
220 REM قيم خلايا مصفوفة سعر التكلفة
230 MAT S = T + R
240 MAT PRINT S
250 REM لإيجاد مصفوفة الاسعار النهائيه
260 FOR I = 1 TO 3
270 FOR J = 1 TO 2
280 LET N(I,J) = INT(S(I,J) + 1) - .01
290 NEXT J
300 NEXT I
310 REM لطباعة سعر البيع للوحدة الواحد
320 PRINT
330 MAT PRINT N
340 DATA 2512,3421,5478,6221,2922,3511
350 DATA 105,110,211,205,100,101
360 DATA .15,.17,.18,.21,.25,.20
370 RETURN

```


وفي حالة تنفيذ البرنامج سنحصل على النتائج التالية :

2512	3421	← مصفوفة التكلفة الكلية
5478	6221	
2922	3511	
105	110	← مصفوفة الوحدات المنتجة
211	205	
100	101	
.15	.17	← مصفوفة نسب الربح
.18	.21	
.25	.2	
23.9238	31.09999	← مصفوفة تكلفة الوحدة
25.96208	30.34633	
29.21999	34.76237	
3.5885	5.2869	← مصفوفة الأرباح
4.6731	6.3727	
7.3049	6.9524	
27.5123	36.3869	← مصفوفة الأسعار للوحدة قبل التحويل إلى ٩٩,
30.6352	36.7190	
36.5249	41.7148	
27.99	36.99	← مصفوفة الأسعار النهائية للوحدة
30.99	36.99	
36.99	41.99	

لاحظ عدم استخدام الرسائل التوضيحية ، لصعوبة تحقيق ذلك على الجهاز الكبير.

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ - بين الأخطاء إن وجدت في كل من العبارات التالية :

- أ - 10 DIM A (5,10), A (2,4), R
ب - 10 MAT READ
ج - 10 PRINT A (I,J); (I,J)
د - 10 FOR I = 9 TO 3
هـ - 10 READ (I,J)

٣ - اعمل على تعديل البرامج التالية لكي يتم تنفيذها بدون أخطاء :

- أ - 10 FOR I = 1 TO 5
20 FOR J = 1 TO 4
30 INPUT A (I,J)
40 PRINT A (I,J);
50 NEXT I
60 PRINT «_____»
70 NEXT J
80 END

- ب - 10 DIM A\$ (4,10)
20 FOR I = 1 TO 4
30 FOR J = 1 TO 5

40 . LET A\$ (I,J) = I

50 NEXT J

60 NEXT I

70 END

10 DIM A (3,2), B (3,2), C (2,2)

— ج —

20 MAT C = A * B

10 DIM A (10), B (5,5) A\$ (20,4)

— د —

20 FOR I = 1 TO 100

30 LET B (I) = I

40 NEXT I

50 END

٤ — اشرح عمل كل من أجزاء البرامج التالية :

110 FOR I = 1 TO 2

— أ —

120 FOR J = 1 TO 4

130 LET A (I) = A (I) + B (I,J)

140 NEXT J

150 NEXT I

110 FOR I = 1 TO 5

— ب —

120 FOR J = 1 TO 7

130 PRINT A\$ (I,J), A (I,J)

140 NEXT J

150 NEXT I

110 DIM A (10), B (10, 10), N\$ (10,10) جـ —

120 MAT INPUT A

130 MAT READ B

140 MAT READ N\$

110 DIM A(2,4), B(2,4), C(4,2), D(4,4), E(5,5) د —

120 MAT C = TRN (N)

130 MAT E = INV (D)

140 MAT PRINT D

٥ — اعمل على تعديل مثال (١١ - ١) في الفصل الحادى عشر ليعمل على قراءة عدد الجرائد المباعة والمعادة في نهاية كل شهر، موزعة على الأسابيع الأربعة وأيام الأسبوع السبعة كما في الجدول التالى :

الأسبوع	الجمعة	الخميس	الأربعاء	الثلاثاء	الاثنين	الأحد	السبت	المجموع
١								
٢								
٣								
٤								
المجموع								المجموع الكلى

المطلوب : القيام بنفس المتطلبات في المثال .

٦ - يعمل ثلاثة أطباء في إحدى العيادات الخاصة حيث يتعامل كل طبيب مع ثلاثة مستشفيات ، ويتم تحويل المراجعين لأحدها حسب طبيعة تشخيص المرض . وفي نهاية كل أسبوع يقوم السكرتير بتحضير قائمة كالتالى :

عدد المرضى المحولين	رقم الطبيب	رقم المستشفى
xxx	«١» أو «٢» أو «٣»	«١» أو «٢» أو «٣»

أ - طور برنامجاً لقراءة البيانات المدخلة في نهاية الأسبوع وتخزينها في مصفوفة ذات بعدين ، ومن ثم حساب مجموع المرضى المحولين لكل طبيب ولكل مستشفى .
أمثلة من البيانات المدخلة :

1, 2, 5

2, 2, 3

1, 3, 4

3, 2, 5

ب - طور برنامجاً لإيجاد رقم الطبيب ورقم المستشفى اللذين لهما أقل وأكبر عدد من المرضى المحولين .

ج - وإذا علم أن دخل الطبيب لكل مريض حسب المستشفى كالتالى :

الدخل بالريالات	رقم المستشفى
٥٠٠٠	١
٧٠٠٠	٢
٦٠٠٠	٣

فأوجد مجموع دخل كل طبيب من كل مستشفى .

الجزء
الثالث

مواضيع متقدمة فى لغة بيك

استخدام الملفات
فى معالجة البيانات
تطبيقات
السلات
أساليب التعامل معها
وتطبيقاتها.
الرسومات البيانية.

استخدام الملفات فى معالجة البيانات

مقدمة عن معالجة الملفات :

تعرضنا فى الفصول السابقة لكيفية إدخال البيانات للحاسب ومعالجتها، ومن ثم الحصول على النتائج إما بإظهارها على الشاشة أو بالحصول على تقارير مطبوعة. ولو أردنا تكرار الحصول على هذه التقارير (نتائج المعالجة) فسنضطر لتشغيل البرنامج مرة أخرى، وفى حالة تغيير البيانات المسندة أو المرفقة مع البرنامج سنضطر لإدخالها مرة أخرى أيضاً. وهذا ما يعرضنا أحياناً لبعض المشاكل، وبذل جهد ووقت إضافيين. وسنتعرض الآن لكيفية الحيلولة دون ذلك باستخدام الملفات.

- فالملف هو عبارة عن (سجل RECORD) أو أكثر من البيانات المترابطة والتي تصف شيئاً معيناً مثل الموظف أو قطع الأثاث.
- والسجل هو عبارة عن (حقل FIELD) أو أكثر من البيانات المترابطة والتي تصف آحاد أو أفراد الشيء الموصوف، مثل سجل موظف معين.
- والحقل هو عبارة عن (حرف / رقم / رمز CHARACTER) أو مجموعة من الحروف والأرقام والرموز، التي تصف حقيقة معينة عن الشيء الموصوف مثل «اسم الدارس».

فمثلاً فى قسم شؤون الموظفين يوجد ملف لجميع موظفى المؤسسة يحتوى على بيانات عن كل موظف منذ فترة التحاقه بالمؤسسة.

فنجده مثلاً : اسم الموظف، رقم الموظف، تاريخ التحاقه بالمؤسسة، المؤهل، الراتب الأساسى، علاوات، اقتطاعات...

فكل واحد من هذه البيانات يعتبر حقلاً FIELD ، ومجموع هذه الحقول يسمى السجل (سجل الموظف) ، ومجموع سجلات الموظفين يطلق عليها اسم الملف .

طرق التعامل مع الملفات :

هناك طريقتان للتعامل مع الملفات : أى كتابة البيانات فى الملفات ، وقراءة البيانات من الملفات ، وتعديل هذه البيانات — وهما :

١ — التعامل بطريقة تتابعية : SEQUENTIAL ACCESS

وهذا يعنى قراءة أو كتابة سجلات الملف بالتسلسل : الأول فالثانى فالثالث وهكذا... إلى نهاية الملف .

ومن مميزات هذا الأسلوب أنه يصلح عند التعامل مع جميع بيانات (سجلات) الملف سواء كان هذا التعامل قراءة أو كتابة . ومن عيوبه عدم المقدرة على تعديل بيانات سجل معين على نفس الملف ، إذ يتطلب ذلك إنشاء ملف جديد تسجل عليه البيانات السابقة ، سواء تلك التى لم تخضع للتعديل أو تلك التى عدلت ، سجلاً سجلاً ، حتى نهاية الملف .

وبعد إنهاء العملية نقوم بحذف الملف القديم وتسمية الملف الجديد باسم الملف القديم . ويمكن استخدام الأشرطة والأسطوانات المغنطة كوسائل لتخزين واسترجاع البيانات بالطريقة التتابعية .

٢ — التعامل بطريقة عشوائية : RANDOM ACCESS

وهى إمكانية التعامل مع أى من سجلات الملف دون المرور على مايسبق هذا السجل من سجلات أخرى . ويتم التعامل مع هذه السجلات عن طريق مفتاح KEY

معين يحدد السجل المراد. وغالباً ما يكون هذا المفتاح عبارة عن أحد الحقول الرئيسية في السجل كرقم الوظيفة مثلاً (حيث يوجد لكل موظف رقم خاص به يختلف عن رقم أى موظف آخر).

فوائد استخدام الملفات :

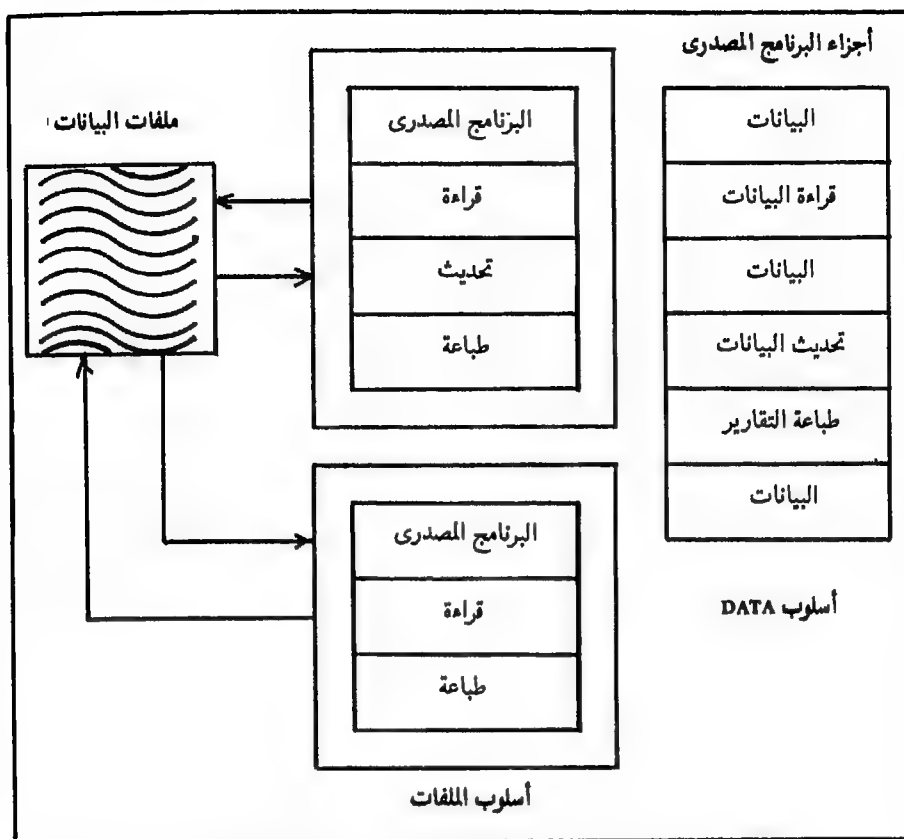
ومن الفوائد الرئيسية لاستخدام الملفات :

(١) تخزين البيانات في إحدى وسائل التخزين المستخدمة، وإمكانية الرجوع إليها في المستقبل حين الحاجة إليها ونقلها من مكان إلى آخر.

(٢) إمكانية استخدام هذه البيانات (الملف) من قبل أكثر من برنامج وذلك بالاستغناء عن تعليمة (بيانات DATA) وتوجيه البرنامج للتعامل مع الملف الذى يحتوى على البيانات المرادة.

(٣) إمكانية الحصول على تقرير مطبوع عن هذه البيانات (الملف) أو إظهار البيانات على الشاشة وقت الحاجة باستخدام إحدى تعليمات أنظمة التشغيل الخاصة بذلك دون الحاجة لكتابة برنامج.

ويبين الشكل التالى الفرق بين أسلوب دمج البيانات فى البرنامج المصدري باستخدام DATA وأسلوب الملفات .



تعليمات معالجة الملفات بالطريقة التتابعية :

وفيما يلي سنتعرض للأشكال العامة لكل من التعليمات الخاصة بتجهيز الملف والتعامل معه ، قراءة وكتابة ، وتحديد الانتهاء منه .

(١) تجهيز وإنشاء الملف (فتح الملف) المتتابع :

الشكل العام لتعليمة فتح الملف						
OPEN المتابع						
X X X	OPEN	« XXXX »	FOR	XXXX	AS	# N
↓	↓	↓	↓	↓	↓	↓
رقم السطر	تعليمة افتح	اسم الملف	لأجل	إحدى التعليمة التالية (INPUT, OUTPUT, APPEND)	كانه	رقم الملف
		↓				
		قواعد اختيار اسم الملف				

وفي حالة تنفيذ هذه العبارة سيتم إنشاء ملف يطلق عليه الاسم الموجود بين علامتي التنصيص، وهذا الإنشاء يتم في حالة تنفيذ هذه العبارة للمرة الأولى، أما في المرات المتتالية فيتم فتح الملف الذى اسمه يتطابق مع اسم الملف الموجود بين علامتي التنصيص، وسيستخدم هذا الملف على حسب التعليمة التى تلى تعليمة (لأجل FOR)، فمثلاً إذا كانت هذه التعليمة تساوى :

- ١ - للإدخال INPUT فذلك يعنى أن هذا الملف سيستخدم للقراءة منه.
- ٢ - للإخراج OUTPUT فذلك يعنى أن هذا الملف سيستخدم للتخزين (كتابة السجلات) فيه لأول مرة.
- ٣ - للإضافة APPEND فذلك يعنى أن هذا الملف يحتوى على بيانات مخزنة ويراد إضافة بيانات (سجلات) جديدة تلى البيانات الموجودة فى الترتيب، ويمكن لهذه التعليمة أن تحل محل عبارة OUTPUT.

أما تعليمة (كأن AS) فنستخدمها لتحديد رقم تسلسل هذا الملف فى البرنامج، فهنا مثلاً تم تحديد رقم الملف بواحد 1* ويستخدم هذا الرقم للتفريق ما بين الملفات (فى

* قواعد اختيار اسم الملف :

- ١ - يجب ألا يزيد عدد الأحرف المكون منها الاسم على تسعة أحرف.
- ٢ - يجب أن يكون الاسم خالياً من الرموز الخاصة.

حالة وجود أكثر من ملف) فكل واحد يأخذ رقماً خاصاً به وعبارة (افتح OPEN) خاصة به ، وعندما يراد استخدام أى من هذه الملفات يتم استخدام هذا الرقم حتى يتم تحديد أى من هذه الملفات هو المقصود.

(٢) التعامل مع هذه الملفات، كما هو مبين في الشكل التالي :

الشكل العام لتعليمة التعامل مع الملف (كتابة PRINT أو قراءة INPUT)				
XXX ↓	XXXX ↓	# N ↓	:	X,(,),X,(,),X,(,),..... ↓
رقم السطر	إحدى التعليمتين PRINT أو اقرأ INPUT	رقم الملف الذى حدد فى عبارة افتح OPEN	فاصلة منقوطة	أسماء الحقول التى يتكون منها السجل مفصولة بفواصل لكتابتها على الملف.

يعتمد تركيب هذه العبارة على عاملين رئيسيين :

١- نوع العملية المراد تنفيذها.

٢- تركيب العبارة التي استخدمت لفتح الملف.

فإذا أردنا التعامل مع الملف للقراءة منه INPUT يجب أن تكون التعليمة المستخدمة بعد تعليمة لأجل FOR في عبارة افتح OPEN تساوي تعليمة إدخال INPUT ، وإلا سيحصل خطأ بسبب عدم تطابق نوع العملية المراد التعامل مع الملف بها مع تركيب عبارة افتح OPEN .

كما يجب أن يكون رقم الملف N ## مطابقاً للرقم المستخدم في عبارة افتح OPEN وكذلك يجب أن تكون أسماء هذه الحقول مطابقة لنوعية البيانات التي يتكون منها السجل.

٣) الانتهاء من التعامل مع الملف : يجب إغلاقه قبل إنهاء عمل البرنامج ، وذلك باستخدام تعليمة أغلق كما هو مبين بالشكل التالى :

الشكل العام لتعليمة أغلق CLOSE		
XXX ↓ رقم السطر	CLOSE ↓ تعليمة أغلق	# N ↓ رقم الملف

٤) يمكن الاستفسار بعد قراءة كل سجل عن نهاية الملف (نهاية البيانات) باستخدام تعليمة نهاية الملف EOF وهى اختصار لـ END OF FILE كما هو مبين فى الشكل التالى :

الشكل العام لعبارة الاستفسار عن نهاية الملف EOF					
XXX ↓ رقم السطر	IF ↓ تعليمة إذا	EOF ↓ نهاية الملف	(N) * ↓ رقم الملف	THEN ↓ إذن انتقل إلى	XXX ↓ رقم التعليمة المراد تنفيذها فى حالة الوصول إلى نهاية الملف

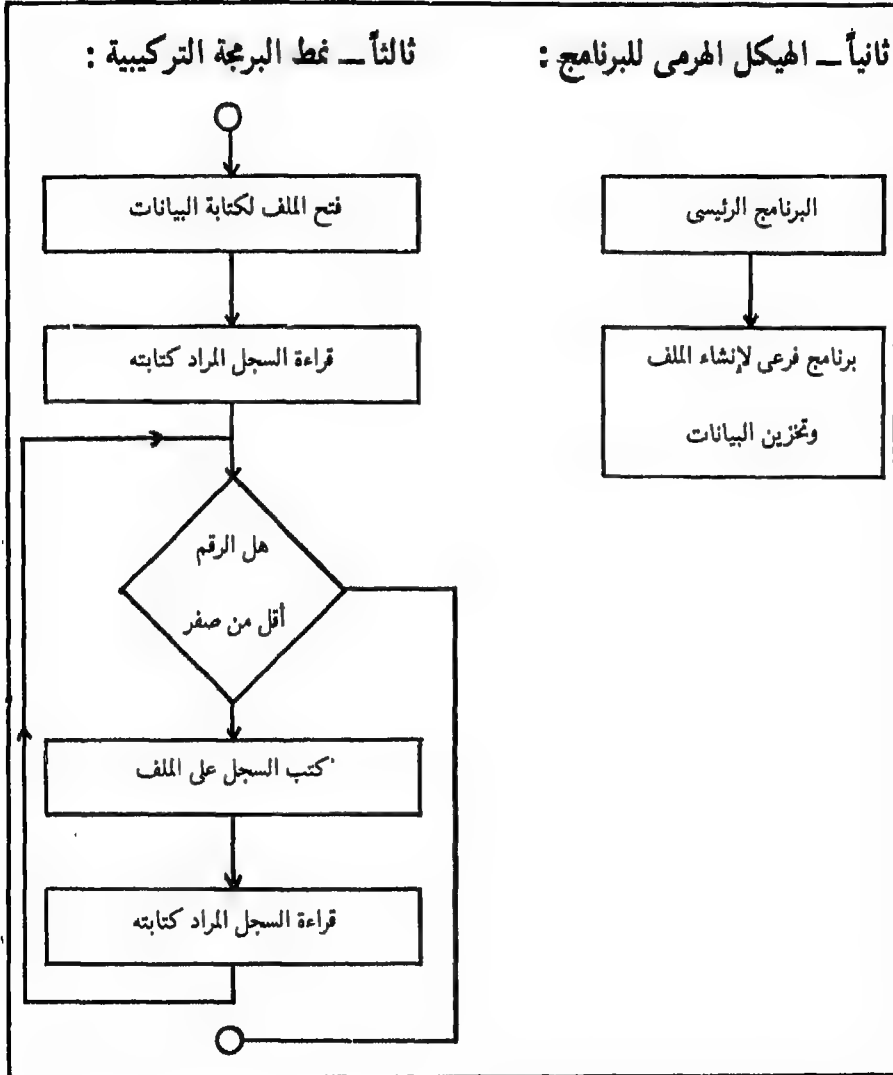
* يجب أن يكون رقم الملف (N) مطابقاً لرقم الملف فى عبارة افتح OPEN وتستخدم هذه التعليمة فى حالة القراءة فقط .

تطوير برامج لاستخدام الملفات بالطريقة التتابعية :

مثال (١٣ - ١) : (الإنشاء والكتابة على الملف)
الهدف : إنشاء ملف لمخزون البضائع فى مستودع رياضى يشمل أرقام البضائع وثمان كل قطعة .

أولاً - خطوات الحل :

- ١ - إنشاء ملف لتخزين البيانات (أرقام القطع وسعر كل منها).
- ٢ - قراءة البيانات المراد تخزينها.
- ٣ - كتابة هذه البيانات على الملف.



رابعاً - وفيما يلي سنتعرض لخطوات حل المثال السابق والعبارات الخاصة بكل خطوة :

خطوة رقم ١ - إنشاء الملف وفتحه ، ويتم ذلك باستخدام تعليمة افتح OPEN
OPEN «PRICES» FOR APPEND AS #1

وفي حالة تنفيذ هذه العبارة سيتم إنشاء الملف وإطلاق الاسم PRICES عليه ،
ولأجل تخزين / كتابة البيانات فيه وإعطائه الرقم ١ .

خطوة رقم ٢ - قراءة البيانات التي ستزود مع البرنامج لتخزينها في الملف ، ويتم
ذلك باستخدام تعليمة اقرأ READ وبيانات DATA

READ N , P

خطوة رقم ٣ - كتابة البيانات / السجلات في الملف ، ويتم ذلك باستخدام تعليمة
اطبع PRINT

PRINT #1; N, P

وعليه سيصبح البرنامج كالتالي :

شكل (١٣ - ١)

برنامج لإنشاء ملف مع بيانات عن مستودع رياضي

```
10 REM استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM برنامج فرعي لتخزين بيانات مستودع في ملف
```

```

-----
50 REM  إنشاء الملف
60 OPEN "prices" FOR APPEND AS #1
70 REM  قراءة رقم القطعة و ثمنها
80 READ N,P
90 REM  استفسار عن نهاية البيانات
100 IF N < 0 THEN 160
110 REM  كتابة البيانات على الملف
120 PRINT #1, N; ", "; P
130 REM  قراءة سجل آخر
140 READ N,P
150 GOTO 100
160 REM  اغلاق الملف
170 CLOSE #1
180 RETURN
190 DATA 801,112.5
200 DATA 704, 56.2
210 DATA 620, 70
220 DATA 512, 48
230 DATA 430, 35
240 DATA 780, 65.5
250 DATA 920, 80.7
260 DATA 815, 70.5
270 DATA 720, 45.5
280 DATA -1,0

```

وبعد تنفيذ البرنامج يتم إنشاء الملف حسب البيانات المعطاة. ويمكن الحصول على نسخة من الملف باستخدام تعليمة : TYPE من وضع نظام التشغيل متبوعة بقائمة البيانات كالتالى :

TYPE	PRICES
801 ,	112.5
704 ,	56.2
620 ,	70
512 ,	48
430 ,	35
780 ,	65.5
920 ,	80.7
815 ,	70.5
720 ,	45.5

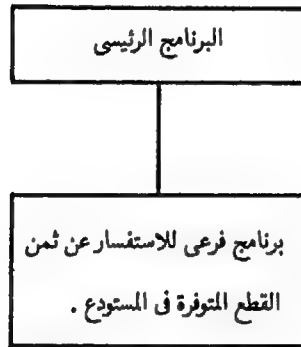
مثال (١٣ - ٢) : (الاستفسار من الملفات)

الهدف : الاستفسار عن أثمان القطع المتوفرة في ملف مخزون البضائع للمستودع الرياضي .

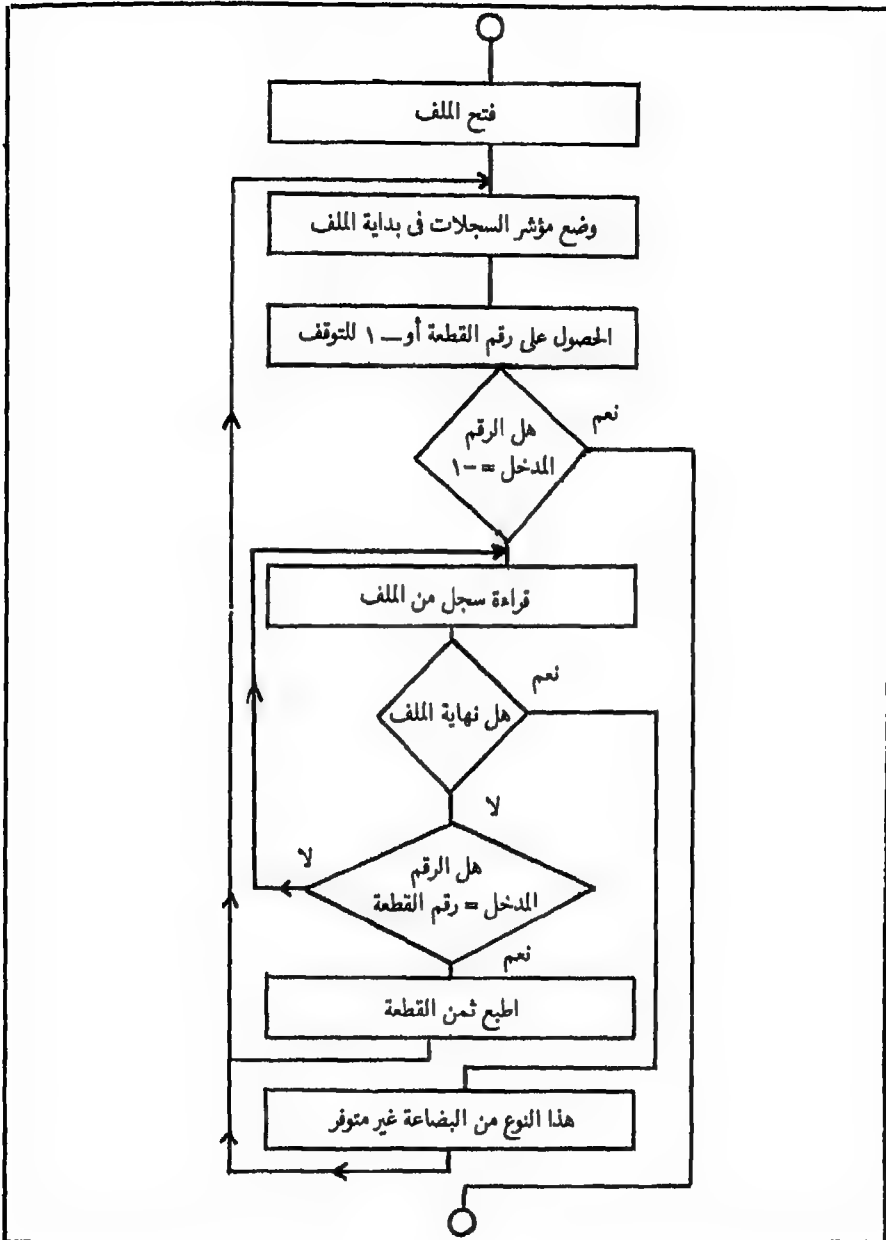
أولاً - خطوات الحل :

- ١ - فتح الملف .
- ٢ - الحصول على رقم القطعة المراد إيجاد ثمنها .
- ٣ - قراءة سجلات الملف والمقارنة بين أرقام القطع المخزنة ورقم القطعة المدخل ، في حالة التطابق عرض أو طباعة ثمن القطعة .

ثانياً - الهيكل الهرمي للبرنامج :



ثالثاً - نمط البرمجة التركيبية :



رابعاً — وفيما يلي سنتعرض لخطوات حل المثال السابق والعبارة الخاصة بكل خطوة :

الخطوة رقم ١ — فتح الملف ويتم ذلك باستخدام التعليمة افتح OPEN

XXX OPEN «PRICES» FOR INPUT AS #1

وفي حالة تنفيذ هذه العبارة سيتم فتح الملف الذى اسمه PRICES والذى تم إنشاؤه فى البرنامج شكل (١٣ — ١) ويحتوى على جميع القطع وثمان كل قطعة ، وتم تحديد الغاية من فتحه لأجل استخدامه للقراءة منه (FOR INPUT) وتم إعطاؤه الرقم ١ .

الخطوة رقم ٢ — الحصول على رقم القطعة المراد الاستفسار عن ثمنها ، ويتم ذلك باستخدام أى من التعليمات السابقة التى تعرفنا عليها فى كيفية إدخال البيانات للمعالجة (اقرأ READ أو أدخل INPUT).

E «أدخل رقم القطعة المراد إيجاد ثمنها أو — ١ للتوقف» ; XXX INPUT

الخطوة رقم ٣ — قراءة سجلات الملف والمقارنة فى كل قراءة إذا كان رقم القطعة المخزن يساوى رقم القطعة المدخل ، اطبع ثمن القطعة ، وإلا فاستمر فى القراءة والمقارنة ، وفى حالة الوصول لنهاية الملف اعرض رسالة للمستخدم بعدم وجود هذه القطعة فى الملف ، ويتم ذلك باستخدام العبارات التالية :

XXX INPUT # 1; N,P

XXX IF EOF (1) THEN (الذهاب لعبارة عرض رسالة عدم وجود هذه القطعة)

XXX IF E = N THEN PRINT P; «ثمان القطعة =»

ELSE GOTO 30

ملاحظة :

يتم إقفال الملف بعد الانتهاء من كل عملية استفسار ، وذلك حتى يتم الرجوع إلى بداية الملف تمهيداً للقيام بالاستفسارات التالية .
وعليه سيكون البرنامج كما هو موضح في الصفحة المقابلة .

وعند تنفيذ البرنامج للاستفسار عن أرقام القطع 620 ، 780 ، 222 ومن ثم التوقف بطباعة 1 — ستكون النتائج كالتالى :

=====

620 ؟ أدخل رقم القطعة المراد إيجاد ثمنها أو- 1 للتوقف

ثمن القطعة = 70

=====

780 ؟ أدخل رقم القطعة المراد إيجاد ثمنها أو- 1 للتوقف

ثمن القطعة = 65.5

=====

222 ؟ أدخل رقم القطعة المراد إيجاد ثمنها أو- 1 للتوقف

هذه القطعة غير موجودة في المستودع

=====

1 - ؟ أدخل رقم القطعة المراد إيجاد ثمنها أو - 1 للتوقف

=====

===== نهاية البرنامج =====

شكل (١٣-٢)

برنامج للاستفسار عن ثمن بضائع مستودع رياضي باستخدام الملفات

```

10 REM استدعاء برنامج فرعي
20 GOSUB 40
30 END
40 REM لفتح الملف
60 OPEN "prices" FOR INPUT AS #1
90 INPUT "ادخل رقم القطعة المراد ايجاد ثمنها أو 1- للتوقف"
100 REM للاستفسار عن نهاية العملية
110 IF E < 0 THEN 230
120 INPUT #1,N,P : REM للحصول على بيانات السجل
130 REM للستفسار عن نهاية الملف
140 IF EOF (1) THEN 200
150 REM للستفسار عما اذا تساوي كل من رقم القطعة
160 IF E <> N THEN GOTO 120
170 PRINT P;" : ثمن القطعة" : CLOSE #1
180 PRINT "===== "GOTO 60
190 GOTO 220
200 PRINT "القطعة غير موجودة في المستودع"
210 PRINT "===== "
220 CLOSE #1: GOTO 50
230 PRINT "===== "
240 RETURN

```

مثال (١٣ - ٣) : (تحديث الملفات) :

الهدف : تعديل أسعار بعض القطع المخزنة في ملف مخزون البضائع للمستودع الرياضى .

أولاً - خطوات الحل :

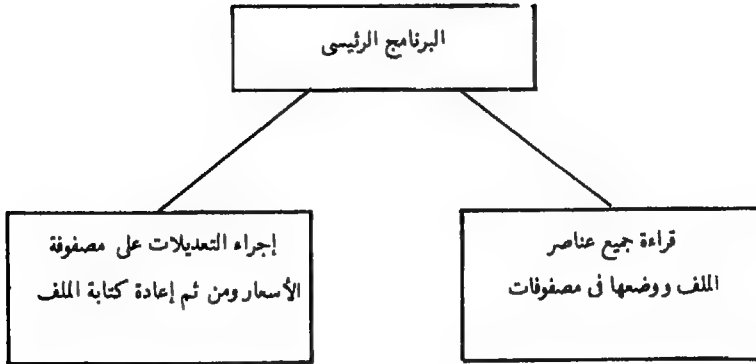
١ - فتح الملف وقراءة جميع السجلات ووضع البيانات في مصفوفات طبقاً لعدد الحقول ومن ثم إغلاق الملف .

٢ - الحصول على رقم القطعة المراد تعديل ثمنها ، وإجراء ذلك على المصفوفات .

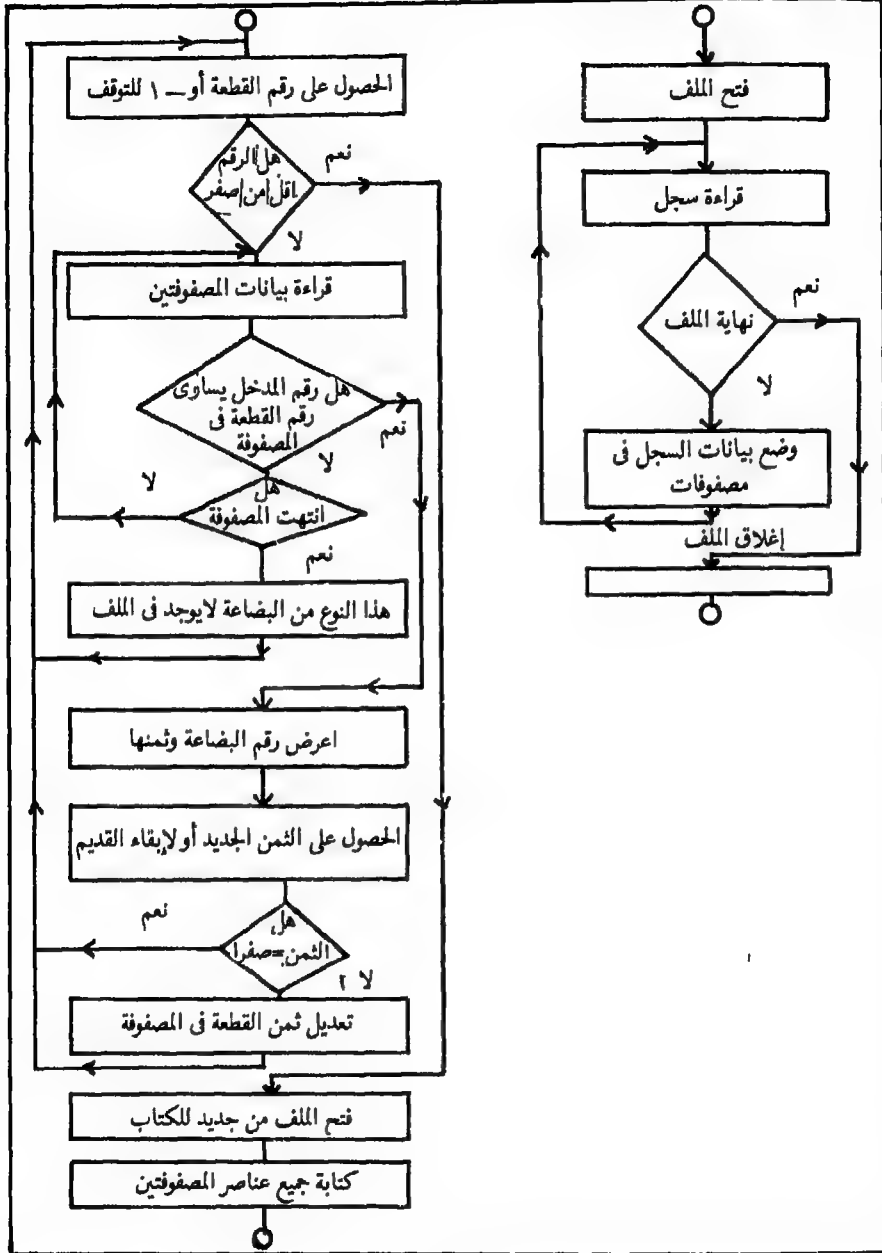
٣ - فتح الملف مرة أخرى وكتابة جميع عناصر المصفوفات على الملف أو إنشاء ملف جديد ، وكتابة جميع عناصر المصفوفات عليه وإعادة تسميته باسم الملف القديم .

وسنتعرض لكليتا الطريقتين في حل المثال .

ثانياً - الهيكل الهرمى للبرنامج :



ثالثاً - مخط البرمجة التركيبية :



رابعاً - البرنامج كاملاً في الشكل التالي عن إعادة كتابة البيانات على نفس الملف :

شكل (١٣-٣)

برنامج لإجراء تعديلات على سجلات الملف المتابع للبضائع

```

10 REM استدعاء برنامج فرعي
20 GOSUB 60
30 REM استدعاء برنامج فرعي
40 GOSUB 140
50 END
60 REM برنامج فرعي لقراءة بيانات الملف و وضعها في مصفوفة
70 DIM N(10),P(10)
80 OPEN "prices" FOR INPUT AS #1
90 FOR I=1 TO 10
100 INPUT #1,N(I),P(I):REM الحصول على بيانات الملف
110 IF EOF(1) THEN 130:REM استفسار عن نهاية الملف
120 NEXT I
130 CLOSE #1:RETURN
140 REM
150 CLS:LOCATE 10,20:PRINT "ادخل رقم المقطع او -1 للوقوف"
160 LOCATE 10,15:INPUT E
170 IF E < 0 THEN 300
180 FOR J = 1 TO I
190 IF N(J) = E THEN 240
200 NEXT J
210 REM
220 LOCATE 12,20:PRINT "الملف " الملف في البضاعة من النوع هذا لا يوجد

```

```

230 GOTO 150
240 LOCATE 12,20:PRINT P(J); "=" الحالي الرقم المقطع N(J) "="
250 LOCATE 14,20:PRINT " لعدد التعديل 0 او الشئ الجديد
260 LOCATE 14,15:INPUT C
270 IF C=0 THEN 150
280 LET P(J)=C
290 GOTO 150
300 REM
310 OPEN "prices" FOR OUTPUT AS #1
320 FOR J=1 TO 1
330 PRINT #1, N(J), "=", P(J):REM كتابة البيانات الجديدة
340 NEXT J
350 RETURN

```

وفي حالة تنفيذ البرنامج لتعديل سعر رقم القطعة 620 من 70 إلى 85 وطباعة محتويات الملف ، ستظهر النتائج كالتالى :

أدخل رقم القطعة أو - 1 للوقوف 620 ؟
رقم القطعة = 620 ثمنها الحالى = 70
أدخل الثمن الجديد أو 0 لعدم التعديل 85 ؟

TYPE	PRICES	
801	,	112.5
704	,	56.2
620	,	85
512	,	48
430	,	35
780	,	65.5
920	,	80.7
815	,	70.5
720	,	45.5

ويمكن تعديل البرنامج السابق لكتابة البيانات على ملف جديد ، ومن ثم إعادة تسمية الملف الجديد وإعطاؤه الاسم القديم وذلك بتعديل العبارات 300 إلى 350 كالتالى :

```
300 REM
310 OPEN "NEWFILE" FOR OUTPUT AS #1
320 FOR J=1 TO I
330 PRINT #1, N(J),",",P(J): REM كتابة السجل على الملف الجديد
340 NEXT J
341 CLOSE #1
342 KILL "PRICES"
345 NAME "NEWFILE" AS "PRICES"
350 RETURN
```

ملاحظات على البرنامج السابق :

١ - كما تلاحظ من شكل (١٣ - ٣) ، تم تعديل الملف PRICES باستخدام المصفوفات ، ومن ثم إعادة كتابتها عليه ، أما إذا أردنا تعديل المصفوفات وكتابتها على ملف جديد ، ومن ثم تسميته باسم الملف القديم ، فيمكن عمل ذلك باتباع الخطوات التالية :

أ) تعديل العبارة رقم (310) لفتح ملف جديد بدلاً من إعادة فتح الملف وذلك بإعطاء اسم آخر للملف .

310 OPEN!« NEWFILE » FOR OUTPUT AS ≠ 1

هنا سيتم إنشاء ملف جديد وإعطاؤه الاسم NEW FILE للكتابة عليه فقط .

ب) تسمية الملف الجديد (NEW FILE) بالاسم القديم (PRICES) ويتم ذلك باستخدام تعليمة (تغيير الاسم NAME) كما هو مبين بالشكل التالي :

الشكل العام لعبارة تعليمة تغيير الاسم NAME				
XXX	NAME	«XXX»	AS	«XXXXXX»
↓	↓	↓	↓	↓
رقم السطر	تعليمة تغيير الاسم	اسم الملف الجديد	مثل / كأنه	اسم الملف المراد تسميته (اسم الملف القديم)

وبناء عليه نضيف العبارة التالية :

345 NAME «NEW FILE» AS «PRICES»

٢ - من المستحسن في معظم حالات التعديل القيام بعمل نسخة من الملف الرئيسى وإجراء التعديلات على النسخة الثانوية ، وتدقيق هذا التعديل في النسخة

الثانوية ؛ للحيلولة دون إجراء أى تعديل خاطيء ، وبعد التأكد من صحة هذا التعديل نقوم بتسمية الملف الجديد الثانوى باسم الملف الرئيسى .
وعملية التدقيق تتم بطباعة أو عرض سجلات الملف ، ويمكننا طباعة هذه السجلات سجلاً بعد الآخر دون التقييد بعدد حقول هذه السجلات بشرط ألا يزيد طول كل سجل على ٢٥٥ حرفاً/رقماً/رمزاً وذلك باستخدام تعليمة (اقرأ سطرًا أو سجلاً LINE INPUT) كما هو مبين بالشكل التالى :

LINE INPUT		الشكل العام لعبارة تعليمة أدخل سطرًا أو سجلاً		
xxx	LINE INPUT	# N	,	x\$
↓	↓	↓	↓	↓
رقم السطر	اقرأ سطرًا/سجلاً	رقم الملف	فاصلة	اسم المتغير الذى سيسند إليه السطر/السجل

• يجب أن يكون اسم المتغير الذى ستسند إليه بيانات السجل غير حسابى . ورقم الملف هو الرقم الذى أسند إليه فى عبارة فتح الملف .

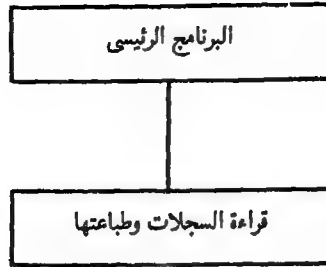
مثال (١٣ - ٤) : (طباعة ملفات)

الهدف : طباعة/ عرض بيانات (سجلات) ملف مخزون البضائع للمستودع الرياضى .

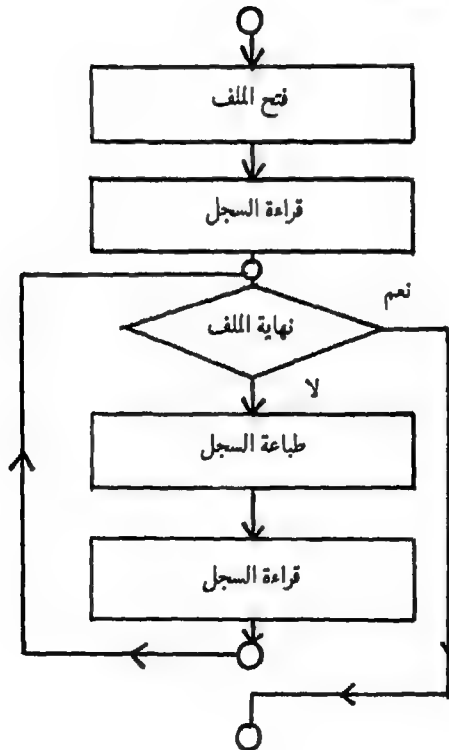
أولاً - خطوات الحل :

- ١ - فتح الملف .
- ٢ - قراءة السجل .
- ٣ - طباعة السجل .

ثانياً - الهيكل الهرمي للبرنامج :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - يحتوي الشكل التالي البرنامج الكامل لهذا المثال :

شكل (١٣-٤)

برنامج لقراءة سجلات ملف وطابعها

```

10 REM
20 GOSUB 40
30 END
40 REM
50 OPEN "PRICES" FOR INPUT AS#1:REM
60 REM
70 CLS:PRINT TAB(30); "
80 PRINT TAB(30); "-----"
90 REM ( قراءة بعد الاخر
100 LINE INPUT #1,A$:REM A$
110 IF EOF(1) THEN 140
120 PRINT TAB(30);A$:REM
130 LINE INPUT #1,A$:GOTO 110
140 PRINT TAB(30);A$:REM
150 PRINT TAB(30); "*****"
160 RETURN

```


وفي حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

```
بيانات ملف مستودع
-----
801 , 112.5
704 , 56.2
620 , 70
512 , 48
430 , 35
780 , 65.5
920 , 80.7
815 , 70.5
720 , 45.5
**** نهاية بيانات الملف ****
```

وبعد مراجعة هذه البيانات (سجلات الملف) وفي حالة عدم الحاجة لها ، يمكن الاستغناء عن الملف (أى حذفه) باستخدام تعليمة (احذف KILL) كما هو مبين في الشكل التالي :

الشكل العام لعبارة تعليمة احذف KILL		
XXX	KILL	«XXXXXX»
↓	↓	↓
رقم السطر	تعليمة احذف	اسم الملف

فمثلا إذا أردنا الاستغناء عن ملف (NEW FILE) في برنامج شكل (١٣ - ٤) نستخدم العبارة التالية :

347 KILL «NEW FILE»

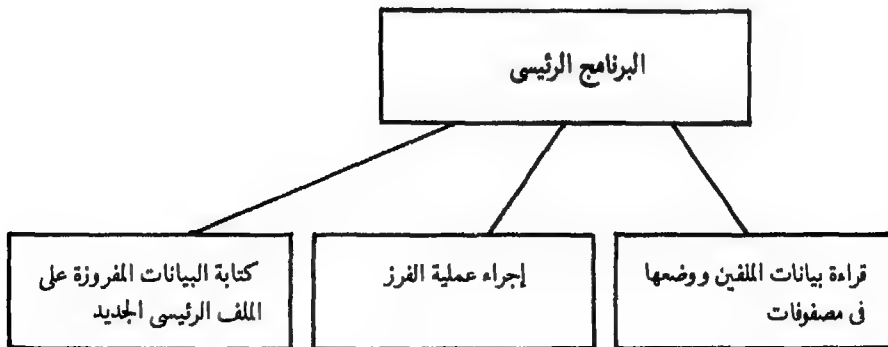
مثال (١٣ - ٥) : (التعامل مع أكثر من ملف) :

الهدف : إضافة بيانات مخزنة على ملف بيانات الملف الرئيسى لمخزون المستودع الرياضى (أى دمجها MERGE) ، ومن ثم فرز هذا الملف الجديد وتخزين البيانات عليه مرتبة ترتيباً تصاعدياً حسب رقم الطباعة (SORTED) .

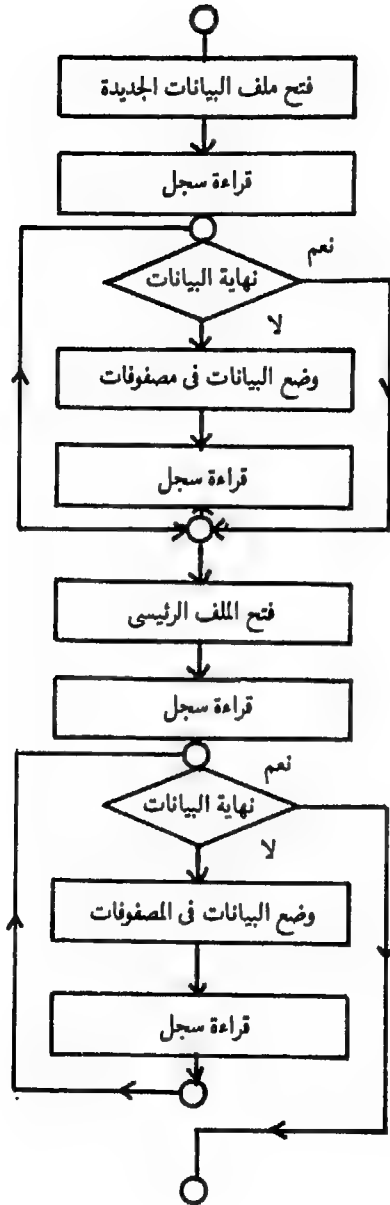
أولاً - خطوات الحل :

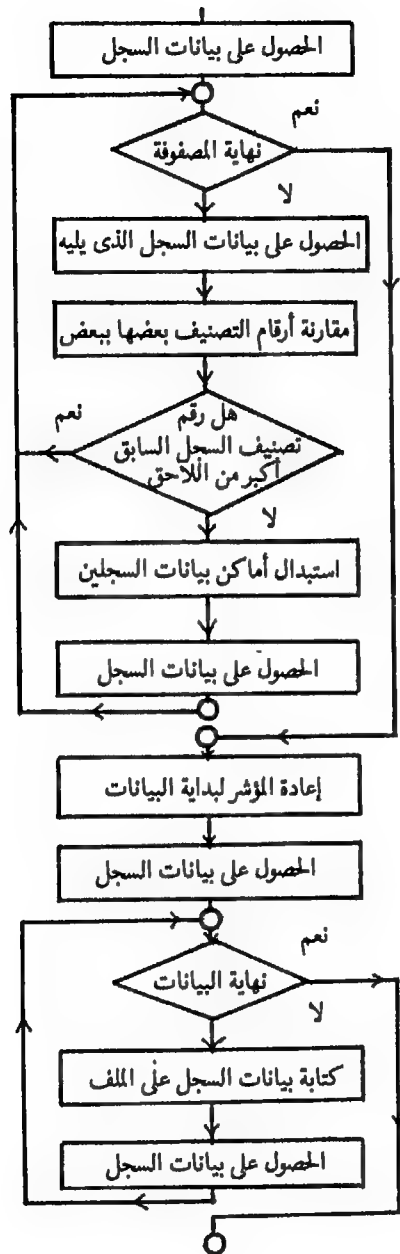
- ١ - فتح ملف البيانات الجديدة TRANSACTIONS .
- ٢ - قراءة هذه البيانات ووضعها فى مصفوفات .
- ٣ - قراءة بيانات الملف الرئيسى وإضافتها على المصفوفات .
- ٤ - إجراء عملية الفرز SORT .
- ٥ - كتابة هذه المصفوفات على الملف الرئيسى الجديد (NEW MASTER FILE) .

ثانياً - الهيكل الهرمى للبرنامج :



ثالثاً - نمط البرمجة التركيبية :





رابعاً - البرنامج في الشكل التالي :

شكل (١٣-٥)

برنامج لدمج ملف بيانات جديدة مع ملف البيانات الرئيسة وفرزها، ومن ثم كتابتها على الملف الرئيس

```

10 GOSUB 50 :REM استدعاء برنامج فرعي
20 GOSUB 200 :REM استدعاء برنامج فرعي
30 GOSUB 310 :REM استدعاء برنامج فرعي
40 END
50 REM "برنامج فرعي لقراءة البيانات الجديدة ووضعها في مصفوفات حسب عدد الحقول"
60 DIM N(100),P(100)
70 REM "فتح ملف البيانات الجديد (TRANS) للقراءة منه"
80 OPEN "TRANS" FOR INPUT AS #1
90 FOR I=1 TO 100
100 INPUT #1,N(I),P(I) :REM مصفوفات في مصفوفات
110 IF EOF (1) THEN 130 :REM تلاستفسار عن نهاية البيانات
120 NEXT I
130 REM "الملف الرئيس وقراءة البيانات ووضعها في مصفوفتي"
140 OPEN "PRICES" FOR INPUT AS #2
150 FOR J=1 TO 100
160 INPUT #2,N(J+1), P(J+1):REM (الدمج)
170 IF EOF (2) THEN 190
180 NEXT J
190 RETURN
200 REM
210 LET R=I+J :REM ايجاد عدد السجلات في الملفين
220 FOR I=1 TO R-1

```

برنامج فرعي لإجراء عملية الفرز على البيانات تصاعدياً

ايجاد عدد السجلات في الملفين

```

230 FOR J=1 TO R-1
240 IF N(J)<= N(J+1) THEN 290
250 REM
260 LET T=N(J):LET N(J)=N(J+1):LET N(J+1)=T
270 REM
280 LET S=P(J):LET P(J)=P(J+1):LET P(J+1)=S
290 NEXT J
300 NEXT I:RETURN
310 REM
320 OPEN "NSORT" FOR OUTPUT AS #3
330 FOR I = 1 TO R
340 PRINT #3,N(I),"",P(I):REM
350 NEXT I
360 REM
365 RESET
370 REM
375 OPEN "TRANS" FOR INPUT AS #1
380 LPRINT TAB(10):"(TRANS (ملف) ادخالها
390 LINE INPUT #1,A$
400 IF EOF(1) THEN LPRINT A$: GOTO 420
410 LPRINT A$:LINE INPUT #1,A$:GOTO 400
420 OPEN "PRICES" FOR INPUT AS #2
425 REM
430 PRINT TAB(10):" (prices (ملف)
440 LINE INPUT #2,A$
450 IF EOF(2) THEN LPRINT A$: GOTO 470
460 LPRINT A$:LINE INPUT #2,A$: GOTO 450
470 OPEN "NSORT" FOR INPUT AS #3

```

تم إيجاد رقم بضاعة أصغر من الرقم الحالي لذا سيتم استبدال الأماكن
بما أنه تم استبدال الأماكن للأرقام لذا سيتم استبدال أسعارها أيضا

برنامج فرعي لكتابة البيانات المدموجة والمفروزة على الملف الرئيسي الجديد
كتابة السجل على الملف الجديد (ملف) ادخالها

وضع المؤشر على بداية ملف البيانات الجديد
البيانات المراد ادخالها

وضع المؤشر على بداية بيانات ملف البيانات القديم
"بيانات الملف الرئيسي القديم (ملف

```

475 REM (newfile ملف )
480 PRINT TAB(10);"(newfile ملف الجديدة
490 LINE INPUT #3, A$
500 IF EOF(3) THEN 520
510 PRINT A$:LINE INPUT #3,A$: GOTO 500
520 PRINT A$:NAME "NEWFILE" AS "PRICES": KILL "NEWFILE"
530 RETURN

```

وفي حالة تنفيذ البرنامج السابق سنحصل على النتائج التالية :

```

      البيانات المراد ادخالها  ملف  (TRANS
777 , 55.5
999 , 66.7
818 , 70.5
735 , 45.5
      بيانات الملف الرئيسي القديم  (ملف  prices
801 , 112.5
704 , 56.2
620 , 70
512 , 48
430 , 35
780 , 65.5
920 , 80.7
815 , 70.5
720 , 45.5
البيانات المدموجه و المفروزه الجديده  (ملف  newfile
430 , 35
512 , 48
620 , 70
704 , 56.2
720 , 45.5
735 , 45.5
777 , 55.5
780 , 65.5
801 , 112.5
815 , 70.5
818 , 70.5
920 , 80.7
999 , 66.7
```


تعليمات معالجة الملفات بالطريقة العشوائية :

توجد هناك اختلافات إلى حد ما في تركيب تعليمات التعامل مع الملفات العشوائية ، عن تركيبها في التعامل مع الملفات التتابعية ، وسنتعرض الآن لكل من هذه التعليمات :

(١) فتح الملف العشوائي : نستخدم تعليمة افتح OPEN كما هو مبين بالشكل التالي :

الشكل العام لعبارة تعليمة افتح OPEN					
XXX	OPEN	«R» ,	N ,	«XXXXXX» ,	N
↓	↓	↓	↓	↓	↓
رقم السطر	تعليمية افتح	للدلالة على أن طريقة التعامل مع الملفات العشوائية	رقم الملف	اسم الملف	طول السجل في الملف

لاحظ الاختلافات في استخدام تعليمة افتح للملفات العشوائية : (١) زيادة «R» للدلالة على أن طريقة التعامل مع الملف هي عشوائية ، و (٢) طول السجل يجب تحديده . وهذا التحديد يعتمد على مجموع طول كل حقل من الحقول التي يتكون منها هذا السجل ، وإذا لم يحدد هذا الطول فسيعتبر الحاسب أن طول كل سجل من سجلات هذا الملف يساوى 255 حرفاً CHARACTER

٢) تحديد طول السجل : لتحديد طول السجل لابد من معرفة طول كل حقل من الحقول التى يتكون منها هذا السجل . ويتم ذلك باستخدام تعليمة الحقل FIELD كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة حقل		FIELD						
XXX	FIELD	≠N,	N	AS	X\$, *	N	AS	X\$,...
↓	↓	↓	↓	↓	↓	↓	↓	↓
رقم السطر	تعليمة تحديد طول كل حقل من الحقول	رقم الملف	طول الحقل	كان	اسم الحقل الأول	طول الحقل	كان	اسم الحقل الثانى

• لاحظ أن جميع أسماء المتغيرات قد حددت كمتغيرات حرفية .

وفى حالة تنفيذ عبارة افتح OPEN فى الملفات العشوائية ، يتم حجز منطقة فى الذاكرة الرئيسية يطلق عليها اسم BUFFER (المنطقة المؤقتة) ، لاستخدامها فى حالة التعامل مع الملف (للقراءة منه أو الكتابة عليه) . فقبل أن تتم الكتابة على الملف الذى تم إنشاؤه على إحدى وسائل التخزين ، يتم وضع جميع محتويات الحقول فى المنطقة المؤقتة BUFFER والتى تمثل السجل . وبعد ذلك تتم كتابة هذا السجل كوحدة واحدة . وما ينطبق على الكتابة ينطبق أيضاً على القراءة من الملف ، ولكن بصورة عكسية . حيث تتم قراءة السجل كوحدة واحدة ووضعه فى المنطقة المؤقتة BUFFER ، ومن ثم يمكننا التعامل مع الحقول حسب مقتضيات التطبيق .

(٣) تجهيز البيانات عقب قراءتها بواسطة تعليمة READ أو INPUT من أجل تخزينها في الملفات العشوائية : وتنطوي هذه الخطوة على نقل البيانات من أسماء المتغيرات التي استخدمت في عمليات الإدخال، وتلك التي استخدمت في تعليمة حقل FIELD . وحيث إن هناك نوعين من البيانات المدخلة — حرفية غير حسابية وعددية حسابية — فيستدعى الأمر اتباع أسلوب خاص لكل منهما .

أ — فالبيانات الحرفية، كالاسم مثلاً في الأجدية اللاتينية، تبدأ كتابتها من اليسار إلى اليمين، ولتخزين (نقل) هذا النوع من البيانات من المتغيرات المستخدمة في عملية الإدخال إلى متغيرات المنطقة المؤقتة BUFFER، تستعمل تعليمة LSET (أى ابدأ بكتابة البيانات في متغير المنطقة المؤقتة من اليسار LEFT JUSTIFIED)، كما هو مبين في الشكل التالى :

الشكل العام لعبارة LSET				
XXX	LSET	X\$	=	X\$
↓	↓	↓	↓	↓
رقم السطر	تعليمية ابدأ الكتابة من اليسار إلى اليمين	اسم المتغير الذى تم استخدامه، في تعليمة الحقول والذى سيستخدم في المنطقة المؤقتة BUFFER	إشارة الإسماد	اسم المتغير الذى سيستخدم في البرنامج للتعامل معه ومن ثم إسناد قيمته للمتغير المساوى له في تعليمة الحقول

ب — أما في حالة قراءة بيانات حسابية ، فيجب القيام بالخطوتين التاليتين لإتمام عملية النقل :

— تحويل القيمة الحسابية إلى قيمة حرفية قبل كتابتها على الملف ، وذلك باستخدام إحدى التعليمات التالية ، وفق الشكل العام التالي :

الشكل العام لعبارة تعليمة تحويل قيمة عددية إلى غير عددية				
XXX	X\$	=	XXXXX\$	X
↓	↓	↓	↓	↓
رقم السطر	اسم المتغير الذي سيستخدم في تعليمة RSET	إشارة الإسناد	إحدى تعليمات التحويل للقيمة العددية إلى قيمة غير عددية	اسم المتغير الذي سيحتوى على القيمة العددية

ويمكن أن تكون تعليمة التحويل إحدى التعليمات التالية :

— MKI\$ وتعمل هذه التعليمة على تحويل القيمة العددية إلى قيمة غير عددية ، وسيتم تمثيلها في ١٦ وحدة ثنائية أى (٢ بايت حيث إن كل بايت يساوى ثمانى وحدات ثنائية) . ويطلق عليها التعبير INTEGER EXPRESSION

— MKS\$ وتعمل هذه التعليمة على تحويل القيمة العددية إلى قيمة غير عددية ، وسيتم تمثيلها في ٣٢ وحدة ثنائية أى ٤ بايت ، ويطلق عليها التعبير SINGLE-PRECISION EXPRESSION

— MKD\$ وتعمل هذه التعليمة على تحويل القيمة العددية إلى قيمة غير عددية ، وسيتم تمثيلها في ٦٤ وحدة أى ٨ بايت . ويطلق عليها التعبير DOUBLE-PRECISION EXPRESSION

— نقل البيانات إلى المنطقة المؤقتة وكتابتها من اليمين إلى اليسار، حيث إنها حسابية، وذلك باستخدام تعليمة RSET (أى ابدأ بكتابة الأرقام من اليمين RIGHT JUSTIFIED) كما هو مبين في الشكل التالى :

الشكل العام لعبارة تعليمة RSET				
XXX	RSET	X\$	=	X\$
↓	↓	↓	↓	↓
رقم السطر	تعليمة ابدأ الكتابة من اليمين إلى اليسار	اسم المتغير الذى تم استخدامه في تعليمة حقل FIELD والذى سيستخدم في المنطقة المؤقتة BUFFER	إشارة الإسناد	اسم المتغير الذى استخدم في تعليمة التحويل

ومن الممكن أن تجمع الخطوتان في عبارة واحدة، كما في الشكل التالى :

الجمع بين عمليتي التحويل من عددى إلى حرفى والنقل إلى المنطقة المؤقتة				
XXX	RSET	X\$ =	XXX\$	(X)
↓	↓	↓	↓	↓
رقم السطر	تعليمة النقل والكتابة من اليمين	اسم المتغير في تعليمة حقل FIELD	إحدى تعليمات التحويل من عددى إلى حرفى	اسم المتغير من البرنامج

٤) **الكتابة على الملف :** بعد تحويل القيم المدخلة وإسنادها إلى الحقول الخاصة بها في المنطقة المؤقتة ، يمكننا كتابتها على الملف باستخدام تعليمة ضع PUT كما هو مبين بالشكل التالي :

الشكل العام لعبارة تعليمة ضع / اكتب PUT			
XXX	PUT	# N	, X
↓	↓	↓	↓
رقم السطر	تعليمة ضع / اكتب	رقم الملف الذى أسند إليه فى تعليمة افتح	اسم المتغير الذى سيحتوى على أرقام سجلات الملف

فإذا كانت القيمة التى يحتوئها متغير أرقام السجلات ٧ ، فإن ذلك يعنى أنه حال تنفيذ هذه العبارة ستم كتابة البيانات التى تحتوئها المنطقة المؤقتة BUFFER في السجل السابع ، على الملف .

٥) **القراءة من الملف :** أما فى حالة القراءة من الملف فتتبع الخطوات التالية :
أ — إحضار السجل من الملف كوحدة واحدة ، وذلك باستخدام تعليمة GET ، كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة أحضر GET			
XXX	GET	# N ,	X
↓	↓	↓	↓
رقم السطر	تعليمة أحضر	رقم الملف الذى أسند إليه فى تعليمة افتح	اسم المتغير الذى سيحتوى على رقم السجل فى الملف

ب — يجب تحويل القيم العددية، والتي تم تخزينها في صيغة قيم غير عددية، إلى صيغتها الأصلية (قيم عددية) قبل التعامل معها في البرنامج باستخدام إحدى تعليمات التحويل، كما هو مبين بالشكل التالي :

الشكل العام لعبارة تعليمة تحويل القيم العددية والتي خزنت كغير عددية إلى أصلها عند القراءة من الملف العشوائى				
XXX	X	=	XXX	(X \$)
↓	↓	↓	↓	↓
رقم السطر	اسم المتغير الذى ستسند له القيمة العددية عند التحويل والذى سيستخدم فى البرنامج	إشارة الإسناد	تعليمة التحويل عند القراءة من الملف	اسم المتغير الذى استخدم فى تعليمة حقل

ويمكن أن تكون تعليمة التحويل إحدى التعليمات التالية :

- CVI وتعمل هذه التعليمة على تحويل القيمة غير العددية (العددية الأصل) والتي تم تخزينها في ١٦ وحدة ثنائية (٢ بايت)، وذلك باستخدام تعليمة \$ MKI، إلى قيمة عددية.
 - CVS تحويل القيمة التي خزنت باستخدام تعليمة \$ MKS إلى قيمتها الأصلية (العددية).
 - CVD تحويل القيمة التي خزنت باستخدام التعليمة \$ MKD إلى قيمتها الأصلية (العددية).
- أما القيم حرفية الأصل والتخزين، فيمكن استخدامها مباشرة في البرنامج بنفس أسماء المتغيرات المستعملة في تعليمة حقل FIELD، أو يمكن إسنادها إلى متغيرات جديدة، وذلك حسب طبيعة المعالجات .

تطوير برنامج لمعالجة الملفات بالطريقة العشوائية :

مثال (١٣ - ٦) : (عمليات إنشاء، إضافة، حذف، تعديل، طباعة)

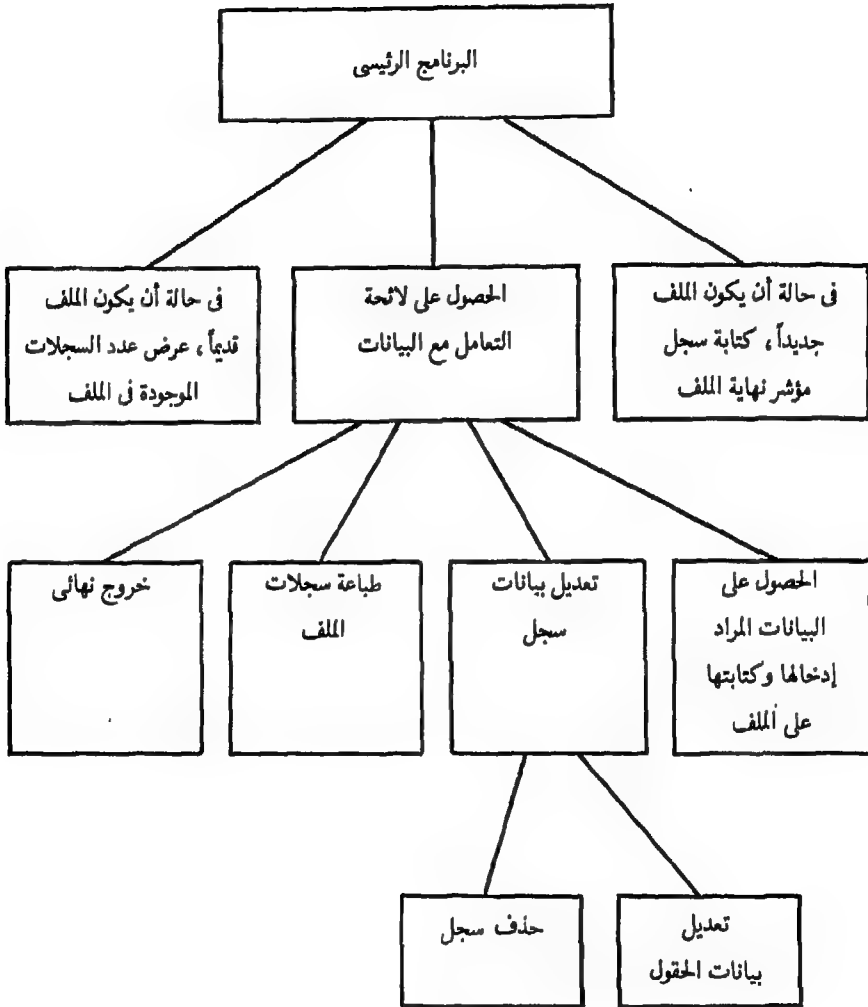
الهدف : إنشاء ملف لمخزون البضائع في مستودع رياضى يشمل جميع أرقام البضائع وثمان كل قطعة، والسماح لمستخدم النظام بالقيام بأى من العمليات التالية :

- أ — إضافة سجل جديد .
- ب — حذف سجل من الملف .
- ج — تعديل بيانات سجل .
- د — طباعة بيانات الملف .

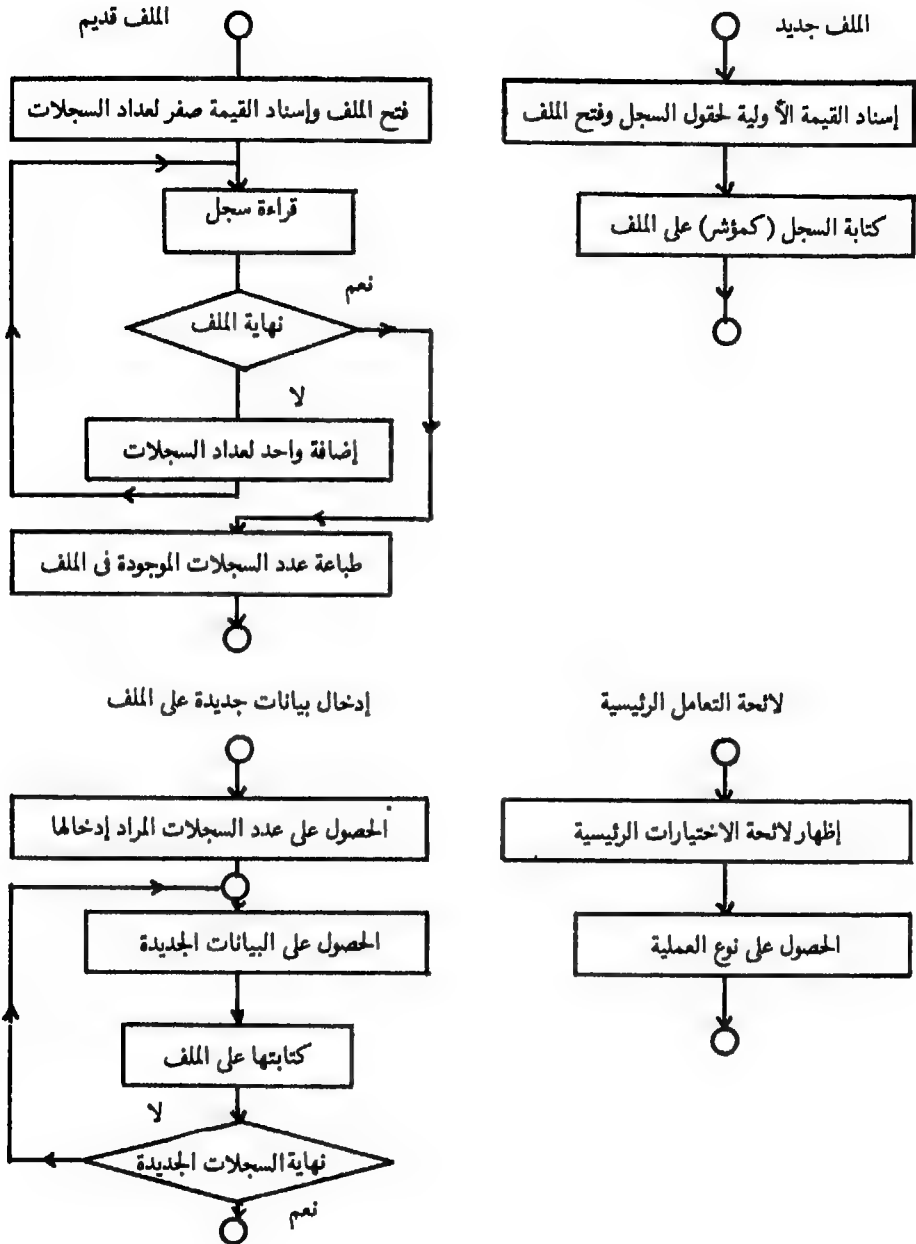
أولاً - خطوات الحل :

- (١) إنشاء الملف وتحديد أطوال الحقول .
- (٢) الحصول على نوع العملية المراد القيام بها .
- (٣) في حالة الإضافة، الحصول على البيانات، ومن ثم كتابتها على الملف .
- (٤) في حالة الحذف، الحصول على السجل المراد حذفه، ومن ثم حذفه من الملف وتعديل الملف .
- (٥) في حالة الحصول على بيانات السجل وعرضها، يتم تحديد الحقل المراد تعديله والحصول على البيانات الجديدة، وإعادة كتابة السجل على الملف .
- (٦) في حالة الطباعة، الحصول على جميع بيانات السجل وطباعتها .

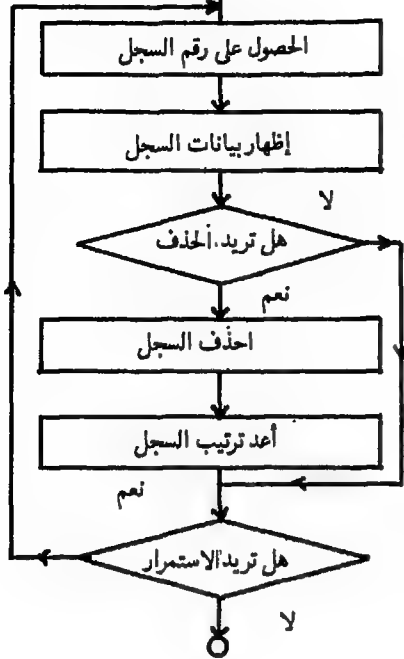
ثانياً - الهيكل الهرمي للبرنامج :



ثالثاً - مخطط البرمجة الهيكلية :



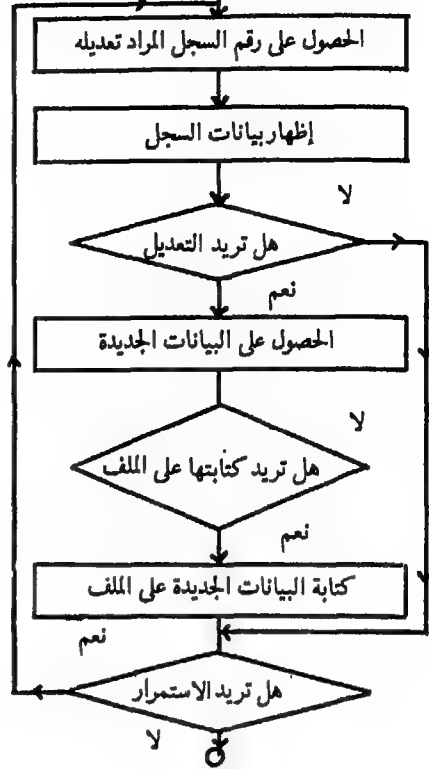
حذف سجل من الملف



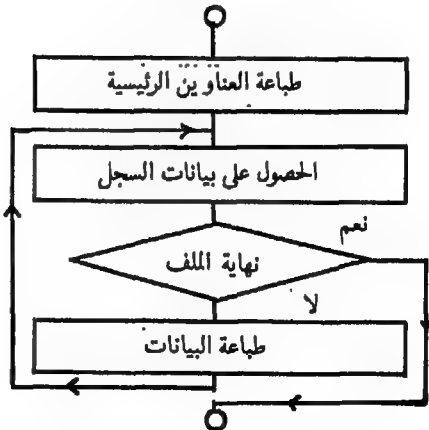
الخروج



تعديل بيانات سجل



طباعة سجلات الملف



رابعاً - البرنامج كاملاً في الشكل التالي :

شكل (١٣-٦)

برنامج لإضافة وتبديل وحذف وطباعة بيانات مستوعج رياضي باستخدام الطريقة المبرمجة

```

10 REM ----- البرنامج الرئيسي -----
20 REM      لفتح الملف العشوائي " RPRIESE " وتحديد طول السجل
30 OPEN "RPRIESE" AS #1 LEN=12
40 REM      لتحديد طول كل من الحقول ( الرقم N$ والسعر P$
50 FIELD #1,6 AS N$, 6 AS P$
60 REM      للا ستفسار هل الملف جديد أم لا
70 CLS: LOCATE 10,20
80 PRINT " هل الملف جديد أم لا أدخل 1 = نعم أو 2 = لا
90 LOCATE 10,20: INPUT A
100 ON A GOSUB 320,200
110 REM      لمراجعة ما تم إدخاله
120 IF A>0 AND A<=2 THEN 140
130 PRINT "=====
140 REM      برنامج فرعي للا تهيئة الاختيارات الرئيسية
150 GOSUB 380
160 ON T GOSUB 570,770,1250,1360
170 IF T <=3 THEN 140
180 CLOSE #1
190 END
200 REM      الملف الموجود في الملف
210 LET R=0:CLS
220 LET R=R+1:REM      إضافة واحد لعداد سجلات الملف

```

```

230 GET #1,R:REM للحصول على سجل
240 REM للا ستفسار عن السجل اذا كان مؤشر نهاية الملف
250 IF CVS(N$) <> 0 THEN 220
260 REM لاظهار عدد سجلات الموجودة على الملف
270 LOCATE 10,25:PRINT R-1;" =
280 LOCATE 15,25:PRINT "اضغط على اي مفتاح للا ستمرار"
290 LET I$=INKEY$
300 IF I$=" " THEN 290
310 RETURN
320 REM برنامج فرعي لا نشاء الملف ووضح سجل لمؤشر نهاية البيانات فيه
330 LET N=0: LSET N$=MK$(P)
340 LET P=0: RSET P$=MK$(P)
350 LET R=1:REM عدد السجلات
360 PUT #1,R:REM كتابة السجل لمؤشر نهاية البيانات على الملف
370 RETURN
380 REM برنامج فرعي لا يظهر الا فترات الرئيسية
390 CLS
400 LOCATE 5,30:PRINT "=====
410 LOCATE 6,30:PRINT " لا فحة الا فترات الرئيسية
420 LOCATE 7,30:PRINT "=====
430 LOCATE 8,30:PRINT " نوع العملية
440 LOCATE 9,30:PRINT "-----الرمز
450 LOCATE 10,30:PRINT "1 اضافة سجل
460 LOCATE 11,30:PRINT "2 تعديل سجل
470 LOCATE 12,30:PRINT "3 طباعة جميع سجلات الملف
480 LOCATE 13,30:PRINT "4 الجزء النهائي
490 LOCATE 14,30:PRINT "-----
500 LOCATE 15,30:PRINT "ادخل رمز العملية المطلوبه :
510 LOCATE 15,30:INPUT T

```

لمراجعة رمز العملية المدخلة

```

520 REM
530 IF T>0 AND T<5 THEN 560
540 LOCATE 20,17: FOR I=1 TO 5: BEEP: NEXT I
550 PRINT "===== في الا دخال "
560 RETURN
570 REM -----
580 REM
590 CLS:LOCATE 5,30:PRINT " برنامج فرعي لاضافة سجل جديد للملف
600 LOCATE 10,25:PRINT " لا دخال بيانات سجل جديد "
610 LOCATE 10,42:INPUT N:LOCATE 10,20:INPUT P
620 REM
630 LSET N$=MKS$(N)
640 LSET P$=MKS$(P)
650 PUT #1,R:LET R=R+1:REM كتابة السجل مع الذاكرة عداد السجلات بواحد
660 GOSUB 690: IF T2=1 THEN 590
670 LET N=0:LSET N$=MKS$(N):LET P=0: LSET P$=MKS$(P)
680 PUT #1,R:RETURN:REM كتابة سجل نهائية البيانات (المؤشر كوالعداد
690 REM
700 REM
710 LOCATE 20,30:PRINT " برنامج فرعي لا
720 LOCATE 20,30:INPUT T2
730 IF T2>=1 AND T2<=2 THEN 760
740 LOCATE 21,30:FOR I=1 TO 5: BEEP:NEXT I
750 PRINT "-----خطا في الا دخال
760 RETURN
770 REM -----
780 REM
790 CLS:LOCATE 3,30:PRINT R-1
800 LOCATE 3,34:PRINT " = "عدد السجلات الموجودة في الملف

```

```

810 LOCATE 5,30 :PRINT "
820 LOCATE 5,30: INPUT R3
830 IF R3>0 AND R3<=R THEN 860 :REM
لمراجعة كلمة رقم السجل المدخل
840 LOCATE 21,20:FOR I=1 TO 5 :BEEP:NEXT I
850 PRINT "-----"
860 GET #1,R3 :REM
للمحول على السجل المراد
870 CLS:LOCATE 5,20:PRINT "
= وثنيتها
880 REM
لتحويل بيانات السجل من القيمه الغير عدديه الى عدديه
890 LET N=CVS(N$):LET P=CVS(P$)
900 LOCATE 5,20 :PRINT P:LOCATE 5,42:PRINT N
910 LOCATE 6,30 :PRINT "
920 LOCATE 7,30 :PRINT "
لا ثمة اختيارات التعديل
930 LOCATE 8,30 :PRINT "
940 LOCATE 9,30 :PRINT "
نوع العملية
950 LOCATE 10,30:PRINT "
الرمز
960 LOCATE 11,30:PRINT "
حذف السجل من الملف
970 LOCATE 12,30:PRINT "
تعديل بيانات السجلات
980 LOCATE 13,30:PRINT "
ابقاء البيانات كما هي
990 LOCATE 14,30:PRINT "
1
2
3
1000 LOCATE 15,30:PRINT "
ادخل رمز العملية
1010 LOCATE 15,30:INPUT T4
1020 IF T4>0 AND T4<=3 THEN 1050 :REM
لمراجعة رمز العملية المدخلة
1030 LOCATE 21,30:FOR I=1 TO 5 :BEEP:NEXT I
1040 PRINT "-----"
1050 IF T4=1 THEN GOSUB 1100 :REM
استدعاء برنامج فرعي لحذف السجل
1060 IF T4=2 THEN GOSUB 1180 :REM
استدعاء برنامج فرعي لتعديل البيانات
1070 GOSUB 690 :REM
العملية
1080 IF T2=1 GOTO 790 :REM
للا ستفسار عن استمرار تنفيذ العملية
1090 RETURN

```

```

1100 REM -----
1110 REM
1120 GET #1,R-1:REM
1130 PUT #1,R3:REM
1140 GET #1,R:REM
1150 PUT #1,R-1:REM
1160 LET R=R-1:REM
1170 RETURN
1180 REM -----
1190 REM
1200 LOCATE 16,30:PRINT "-----"
1210 LOCATE 18,25:PRINT "      "
1220 LOCATE 18,25:INPUT P:LOCATE 18,41:INPUT N
1230 LSET N#=MKS$(N):LSET P#=MKS$(P):PUT #1,R3
1240 RETURN
1250 REM -----
1260 REM
1270 CLS:FOR I=1 TO R-1
1280 GET #1,I
1290 LOCATE I,10:PRINT "      "
1300 LET N=CVS(N$):LET P=CVS(P$)
1310 LOCATE I,10:PRINT P:LOCATE I,26:PRINT N
1320 LOCATE I,50:PRINT I
1330 FOR W=1 TO 8000:NEXT W:REM
1340 NEXT I
1350 RETURN
1360 REM -----

```

برنامج فرعي لحذف السجل من الملف
للحصول على آخر سجل تم ادخاله
وضع المؤشر في السجل المراد
للحصول على آخر سجل (المؤشر)
تقديمه بمركز واحد
طرح واحد عن عدد سجلات الملف

برنامج فرعي لتعديل حقول السجل
"ادخل رقم البضاعه
وثنيتها

برنامج فرعي لطباعة محتويات الملف

السجل : رقم البضاعه : ثمنيتها :

للتاخير في تنفيذ البرنامج حتى تنتهي القراءه


```

1370 REM
1380 CLS:FOR I=30 TO 50:LOCATE 7,1:PRINT "=";NEXT I
1390 FOR I=7 TO 14
1400 LOCATE I,30:PRINT "=":LOCATE I,50:PRINT "=":NEXT I
1410 FOR I=30 TO 50:LOCATE 14,1:PRINT "=";NEXT I
1420 LOCATE 11,31:PRINT "      جمع السلاحه"
1430 FOR I = 1 TO 5000:NEXT I:REM      تنتم القراءه
1440 RETURN

```

برنامم فرعم لطبامه رساله مع السلاحه

وفي حالة تنفيذ البرنامج سنحصل على النتائج التالية :

أ — تقريراً ما إذا كان السجل جديداً، أى يكتب عليه البيانات لأول مرة .

هل الملف جديد أم لا أدخل 1 = نعم أو 2 = لا

أما إذا أدخل «2» فيتم طبع عدد السجلات في الملف ومن ثم الاستمرار في التنفيذ .

ب — إظهار لائحة الاختيارات الرئيسية :

لائحة الاختيارات الرئيسية	
الرمز	نوع العملية
1	إضافة سجل جديد للملف
2	تعديل سجل
3	طباعة جميع سجلات الملف
4	الجزء النهائي

أدخل رمز العملية المطلوبة : 1 ؟

ج — إدخال البيانات (ثلاثة سجلات)

لإدخال بيانات سجل جديد

أدخل رقم القطعة : 122 ؟ وثمانها : 33 ؟

هل تريد الاستمرار أدخل 1 = نعم 2 = لا 1 ؟

لإدخال بيانات سجل جديد

أدخل رقم القطعة : 635 ؟ وثمنها : 45 ؟

هل تريد الاستمرار أدخل 1 = نعم 2 = لا 1 ؟

لإدخال بيانات سجل جديد

أدخل رقم القطعة : 817 ؟ وثمنها : 61 ؟

هل تريد الاستمرار أدخل 1 = نعم 2 = لا 2 ؟

د - لطباعة محتويات الملف عن طريق الاختيار رقم «3» في الشاشة الرئيسية :

رقم السجل :	1	رقم البضاعة :	122	ثمنها :	33
رقم السجل :	2	رقم البضاعة :	635	ثمنها :	45
رقم السجل :	3	رقم البضاعة :	817	ثمنها :	61

هـ - تعديل سجلات الملف عن طريق إدخال الاختيار رقم «2» في الشاشة الرئيسية :

عدد السجلات الموجودة في الملف = 3

أدخل رقم السجل المراد تعديل بياناته 2 ؟

و - إظهار رقم القطعة و ثمنها للبدء في عملية التعديل ، إظهار لائحة اختيارات التعديل ، ومن ثم إدخال البيانات الجديدة :

و ثمنها = 45

رقم القطعة = 635

لائحة اختيارات التعديل	
الرمز	نوع العملية
1	حذف السجل من الملف
2	تعديل بيانات السجلات
3	إبقاء البيانات كما هي

أدخل رمز العملية 2 ؟

أدخل رقم البضاعة 824 ؟ و ثمنها 73 ؟

هل تريد الاستمرار أدخل 1 = نعم 2 = لا 2 ؟

ز — العودة إلى الشاشة الرئيسية وطباعة سجلات الملف بعد التعديل :

رقم السجل : 1 رقم البضاعة : 122 ثمنها : 33
رقم السجل : 2 رقم البضاعة : 824 ثمنها : 73
رقم السجل : 3 رقم البضاعة : 817 ثمنها : 61

ح — حذف ملف وذلك بطباعة رقم الاختيار «2» من الشاشة الرئيسية، وتحديد رقم السجل المراد حذفه، ويتم ذلك من شاشة اختيارات التعديل :

رقم القطعة = 824 وثمنها = 73

لائحة اختيارات التعديل	
الرمز	نوع العملية
1	حذف السجل من الملف
2	تعديل بيانات السجلات
3	إبقاء البيانات كما هي

أدخل رمز العملية 1 ؟

هل تريد الاستمرار أدخل 1 = نعم 2 = لا ؟

ط — طباعة سجلات الملف بعد الحذف :

رقم السجل : 1 رقم البضاعة : 122 ثمنها : 33

رقم السجل : 2 رقم البضاعة : 817 ثمنها : 61

ي — للانهاء، أدخل الاختيار رقم « » من الشاشة الرئيسية :

مع السلامة

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ - بين الأخطاء إن وجدت في كل من العبارات التالية :

- | | | | |
|----|----------------------------|----|------|
| 10 | OPEN "INDATA" FOR INPUT AS | 1 | أ - |
| 10 | LET A\$ = A | | ب - |
| 10 | CLOSE INPUT | | ج - |
| 10 | FIELD # 2, B AS | 4 | د - |
| 10 | OPEN R , # 1, "DATA2" | 20 | هـ - |
| 10 | LET B = MKI\$ (B) | | و - |
| 10 | LET D\$ = CVI (D) | | ز - |

٣ - اعمل على تعديل البرنامج التالى لكى يتم تنفيذه بدون أخطاء ، مع العلم بأن الملف يحتوى على بيانات عن الدارسين (رقم الدارس واسمه فقط) .

أ - بالطريقة التتابعية

- | | | |
|----|-----------------------|------------------------|
| 10 | CLS | |
| 20 | LOCATE 5,80: PRINT " | " 1- للكتابة على الملف |
| 30 | LOCATE 6, 80: PRINT " | " 2- للقراءة من الملف |
| 40 | LOCATE 7, 80: PRINT " | " 3- لإنهاء العمليات |
| 50 | LOCATE 8, 80: PRINT " | -----" |
| 60 | LOCATE 10,81: INPUT A | |
| 70 | ON A GOSB 100,200,90 | |
| 80 | GOTO 10 | |

```

90  END
100 INPUT  "                                N$ و "أدخل رقم الدارس
110 INPUT  "                                N و "أدخل اسم الدارس
120 OPEN  "SDATA" FOR INPUT  AS  1
130 PRINT  1, N, ", " N$
140 CLOSE
150 RETURN
200 CLS
210 OPEN  "SDATA" FOR APPEND
220 READ  ≠  1; N, N$
230 IF EOF  THEN  260
240 PRINT  N$;                                رقم الدارس «N» الاسم :
250 GOTO  220
260 CLOSE  1
270 RETURN

```

ب — بعد إجراء التعديل وإتمام تنفيذ البرنامج السابق (أ) ، اعمل على تطويره للتعامل مع الملف بالطريقة العشوائية .

٤ — يتابع أحد المستثمرين أسعار أسهم شركة ك ل م ، وفي نهاية كل يوم يدخل السعر الجديد ويحسب نسبة التغير السلبى أو الإيجابى من سعر اليوم السابق ، وفي نهاية الشهر يقوم بطبع قائمة بالأسعار بالأيام .

طور برنامجاً لهذا الغرض باستخدام الملفات ، وأظهر النتائج بالأساليب التالية :
— قائمة بالأسعار ونسبة التغير فى كل يوم .

ملاحظة : سيحتوى الملف على حقلى السعر ونسبة التغير .

٥ - يواجه محل السليمان لبيع آلات التصوير مشكلة سببها عدم حفظ سجلات دقيقة تحتوى على الأسعار الحالية للآلات، وقد أدى ذلك إلى خسارة الزبائن بسبب بيع الآلات بأسعار أعلى من المنافسين، أو إلى تحقيق خسارة بسبب بيع الآلات بسعر أقل من سعرها الجديد .

لذلك كان من الضروري الحفاظ على ملف للأسعار الحالية وتحديثه باستمرار، وكذلك للاستفسار عن سعر أية قطعة .

طور برنامجاً ذا ملف عشوائى للأسعار، فيه إنشاء الملف والتحديث والاستفسار .

الفصل الرابع عشر

تطبيقات

شرحنا في الفصول السابقة كيفية تطوير برامج لتطبيقات محددة، وذلك من خلال الأمثلة المختلفة، كان آخرها في الفصل السابق الخاص بالملفات .

وفي هذا الفصل، سنورد عدة تطبيقات ذات طابع عام وشمولي، منها العلمي والتجاري والتعليمي، وستتضمن هذه التطبيقات المجالات التالية :

- ١ - الفرز والدمج والبحث .
- ٢ - معالجة الملفات في تطبيق علامات الدارسين الحكوميين .
- ٣ - استخدام الرسومات في تطبيق تعليمي لعرض أجزاء الحاسب الآلي .

الفرز والدمج والبحث

Sort, Merge, and Search

تعتبر عمليات الفرز والدمج من أكثر العمليات استخداماً في التطبيقات سواء كانت علمية أو تجارية، فنجد أن معظم أقسام المؤسسة تطلب من إدارة مركز الحاسب الحصول على تقارير مطبوعة ومرتبطة ترتيباً تصاعدياً، كتنقيح عن أسماء الموظفين، أو تنازلياً حسب الرواتب .

وسيتتم الشرح التفصيلي للخطوات المتبعة في أساليب الفرز والبحث وإعطاء البرامج الخاصة بكل أسلوب . ومن ثم سيطور برنامج متكامل باستخدام أساليب الفرز والبحث والدمج .

الفرز: هناك العديد من النظريات المستخدمة في عمليات الفرز ولكل منها طريقة تختلف عن الأخرى، وسنتعرض في هذا الفصل لثلاثة من هذه الأساليب ولنزايا

وعيوب استخدام كل منها . وهذه الأساليب هي :

١ - أسلوب الفرز «الفقاعي» Bubble Sort

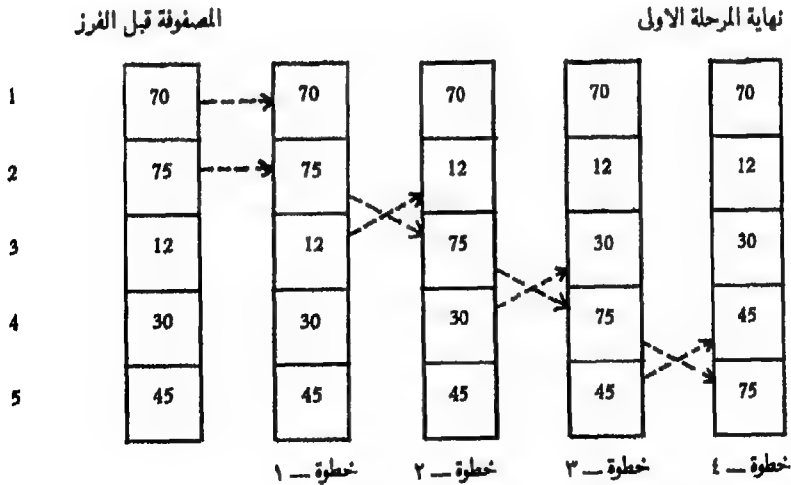
٢ - أسلوب الفرز «الشجري» Heap Sort

٣ - أسلوب الفرز «شل» Shell Sort

مثال (١٤ - ١) : (أسلوب الفقاعات Bubble Sort)

باستخدام هذه النظرية تتم مقارنة كل قيمتين متجاورتين حسب نوع الفرز المراد، فإذا كان الفرز تصاعدياً وكانت نتيجة المقارنة أن القيمة الأولى أصغر من القيمة المجاورة، لن يتم الاستبدال والعكس صحيح في كلتا الحالتين :

فإذا كانت لدينا المصفوفة A التالية فإن عملية الفرز التصاعدي تتم بها كالتالي :



لاحظ أنه في نهاية المرحلة الأولى تم وضع أكبر قيمة من قيم المصفوفة في مكانها الصحيح وهي القيمة (75) والتي يجب أن تكون في الخلية رقم (5)

رقم الخطوة	الخلايا التي قورنت	قيمتها	النتائج	الحدث
١	A (1) مع A (2)	70 مع 75	قيمة الخلية رقم ٢ أكبر	لا يحدث استبدال
٢	A (2) مع A (3)	75 مع 12	قيمة الخلية رقم ٢ أكبر	يحدث استبدال
٣	A (3) مع A (4)	75 مع 30	قيمة الخلية رقم ٣ أكبر	يحدث استبدال
٤	A (4) مع A (5)	75 مع 45	قيمة الخلية رقم ٤ أكبر	يحدث استبدال

لقد احتجنا إلى أربع خطوات في المصفوفة التي عدد خلاياها يساوى خمساً لوضع قيمة واحدة في موضعها الصحيح . ولوضع القيمة الثانية في الموضع الصحيح لها نبحث في أربع خلايا بدلاً من خمس ، وهكذا إلى أن يتم وضع جميع عناصر المصفوفة في الترتيب المطلوب .

والبرنامج التالى يبين هذا الأسلوب في الفرز .

شكل (١٤-١)

برنامج للحصول على أرقام عشوائية باستخدام دالة RND وفرزها باستخدام أسلوب BUBBLE

```

10 DIM A(1000)
20 GOSUB 50:REM عشوائية رقم عشوائي
30 GOSUB 140:REM عملية الفرز
40 END
50 REM rnd دالة
60 CLS:LOCATE 10,20:PRINT "
ادخل عدد الارقام المراد الحصول عليها :
70 LOCATE 10,20 :INPUT N
80 FOR I=1 TO N
90 LET B =RND
100 REM 1000,1 ما بين
110 LET A(I)=INT(B*1000)
120 NEXT I
130 RETURN
140 REM BUBBLE نظرية
150 LET S$=TIME$:REM
لتحديد بداية وقت الفرز
160 LET M=N:REM
اسناد عدد الارقام للمتغير M
170 LET P=0:REM
اسناد القيمة صفر لمؤشر الفرز
180 FOR I=1 TO M-1
190 REM
لاجراء عملية المقارنة بين القيم
200 IF A(I+1)>A(I) THEN 260
210 REM
لم يتحقق الشرط فستتم عملية تفتير اماكن القيم

```

```

220 LET T=A(I+1)
230 LET A(I+1)=A(I)
240 LET A(I)=T
250 REM
260 LET P=I
270 NEXT I
280 REM
290 IF P=0 THEN 330
300 REM (الفرد)
310 LET M=P
320 GOTO 170
330 REM للفرد
340 LET F$=TIME$
350 LOCATE 13,20
360 PRINT "N:"
370 LOCATE 15,20:PRINT F$
380 RETURN

```

يعتبر أسلوب الفرز بالطريقة الفقاعية من أكثر الأساليب ملائمة لفرز البيانات قليلة العدد، ومن أسهل الأساليب تفهماً واستيعاباً خاصة للمبتدئين، لكن في بعض الحالات نحتاج لاستخدام أساليب أخرى في عملية الفرز، حتى ولو احتجنا إلى كتابة عبارات أكثر مما احتجنا في أسلوب الفرز الفقاعي .

مثال (١٤ - ٢) : (أسلوب شل Shell)، و يطلق عليه اسم Shell نسبة إلى مكتشفه D. L. Shell في عام ١٩٥٩م .

والخطوات التي يمر بها هذا الأسلوب كالتالي :

- ١ - قراءة البيانات ووضعها في مصفوفة والحصول على عدد العناصر N
- ٢ - قسمة المصفوفة إلى نصفين $M = \text{INT} (N / 2)$
- ٣ - الاستفسار إذا كان المنتصف يساوى صفراً أى أنه قد تم فرز المصفوفة، وإلا تنفذ الخطوات من ٤ إلى ٦ .
- ٤ - تحديد بداية النصف وذلك بطرح المنتصف (M) من عدد عناصر المصفوفة (N) .

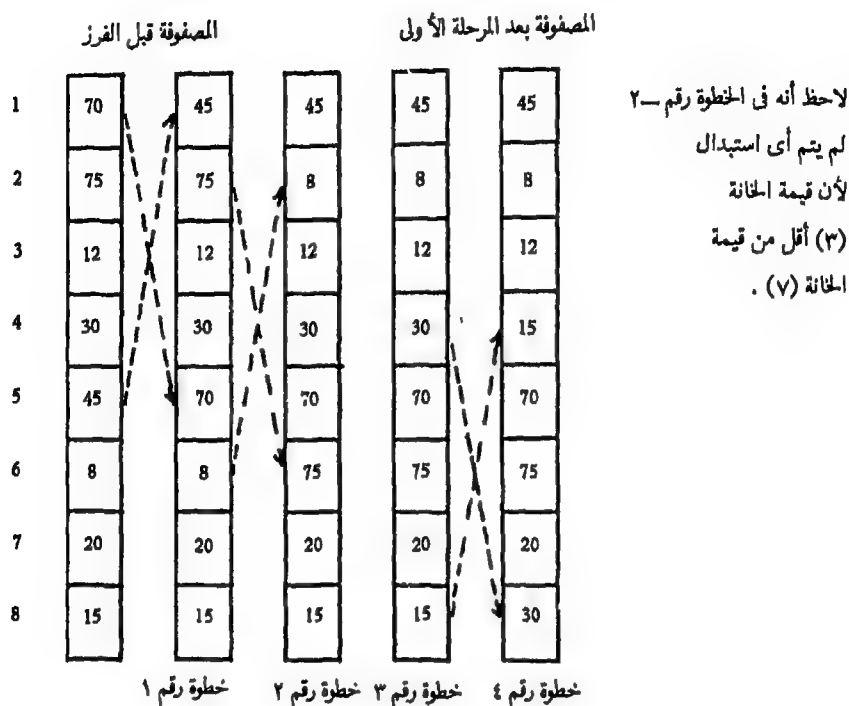
$$S = N - M$$
- ٥ - تحديد نهاية النصف .
- ٦ - العودة للخطوة رقم ٢ .

وخطوات الحل باستخدام هذا الأسلوب تتم بقسمة المصفوفة إلى قسمين، وذلك بالحصول على ناتج (العدد الصحيح) من قسمة عدد خلايا المصفوفة على اثنين، ويمثل هذا الناتج المسافة (عدد الخانات) بين كل خائتين ستم المقارنة بين قيمهما، و يطلق عليه اسم الفراغ .

(1) A High-Speed Sorting Algorithm, Communications of the ACM, July 1959, Vol. 2, PP. 30-32.

فإذا كانت لدينا المصفوفة A التالية فإن عملية الفرز التصاعدي تتم بها كالتالى :

بما أن عدد خلايا المصفوفة = ٨ فإن الفراغ = ٤



فى المرحلة الثانية يتم قسمة الفراغ منتصف المصفوفة إلى نصفين ليمثل الفراغ الذى
سيستخدم فى المرحلة الثانية .

$$\text{الفراغ} = ٢ + ٢ = ٤$$

أى مقارنة قيمة الخانة الأولى بقيمة الخانة الثالثة ، وقيمة الخانة الثانية بقيمة الخانة
الرابعة ، وهكذا ...

المصفوفة بعد المرحلة الأولى من الفرز

	↓ 2		↓ 4		↓ 6		↓ 8
45	8	12	15	70	75	20	30
↑ 1		↑ 3		↑ 5		↑ 7	

وبعد انتهاء المرحلة الثانية من الفرز (المقارنة) نجد أن المصفوفة ستكون كالتالي :

12	8	45	15	20	30	70	75
----	---	----	----	----	----	----	----

وفي المرحلة الثالثة سيكون الفراغ $2 + 2 = 4$ أى مقارنة كل قيمة بالقيمة التي تليها ، وعليه سيكون ناتج المقارنة كالتالي :

8	12	15	20	30	45	70	75
---	----	----	----	----	----	----	----

والبرنامج التالى يبين خطوات هذا الأسلوب فى الفرز .

شكل (١٤-٢)

برنامج للحصول على أرقام عشوائية باستخدام دالة RND ووزنها باستخدام أسلوب SHELL

```

10 DIM A(100).
20 GOSUB 50:REM      استبعاد برنامج فرعي للحصول على الأرقام العشوائية
30 GOSUB 120 :REM      استبعاد برنامج فرعي للأجراء عملية الفرز
40 END
50 CLS
60 LOCATE 10,20: INPUT "ادخل عدد الأرقام المراد فرزها " N
70 FOR I=1 TO N:REM      للحصول على أرقام عشوائية
80 LET B = RND
90 LET A(I) = INT(B*100)
100 NEXT I
110 RETURN
120 REM shell نظرية باستخدام
130 LET S$ = TIME$:REM      للحصول على بداية الوقت
140 LET M = 1
150 LET M = M * 2
160 IF M < N THEN 150
170 LET M = INT((M-1) / 2):REM      لليجاد منتصف المصفوفة
180 IF M = 0 THEN 310
190 LET S = N-M:REM      لتحديد البداية
200 FOR I = 1 TO S
210 LET J = I
220 LET E = J + M:REM      لتحديد النهاية

```

```

230 IF A(E) >= A(J) THEN 290:REM لا جراء المقارنة بين القيم
240 LET T = A(J):REM لم يتحقق الشرط فيتم تغيير اماكن القيم
250 LET A(J) = A(E)
260 LET A(E) = T
270 LET J = J - M:REM طرح عدد ما تم فرزها
280 IF J > 0 THEN 220
290 NEXT I
300 GOTO 170
310 LET F$=TIME$
320 LOCATE 13,20
330 PRINT " N:" من الارقام
340 PRINT " F:" الى shell تم فرد "N:" من الارقام
350 RETURN

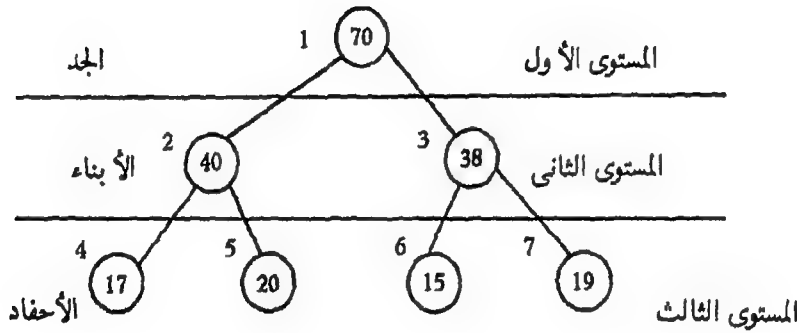
```

مثال (١٤ - ٣) : (الأسلوب الشجري HEAP) :

إن خطوات الفرز بهذا الأسلوب تشابه عملية رسم شجرة العائلة حيث تبدأ بالجد فالأبناء فالأحفاد، والمثال التالي يبين كيفية عمل ذلك . فإذا كان لدينا عائلة مكونة من سبعة أفراد وأعمارهم كالتالي :

19 , 15, 20, 17, 38, 40, 70

فإن شجرة العائلة ستكون

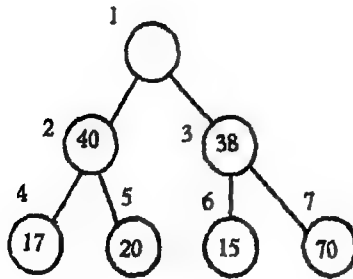


فخطوات الفرز تتم :

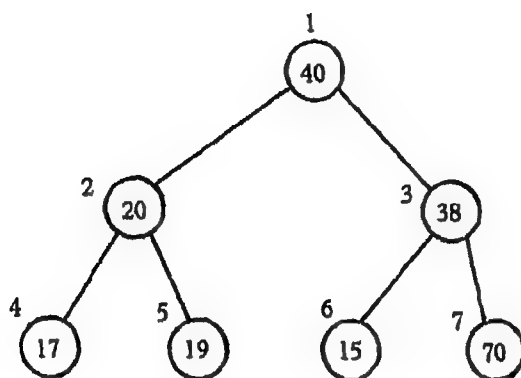
١) بإيجاد أكبر قيمة ووضعها في آخر دائرة، وفي مثالنا هذا آخر دائرة (خلية) هي رقم

سبعة وأكبر قيمة هي 70

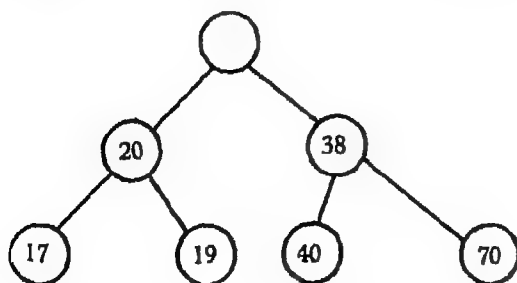
وعليه ستصبح الشجرة كالتالي :



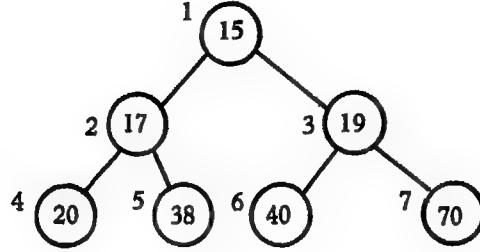
(٢) ومن ثم نحتفظ بالقيمة (19) ونبحث عن القيمة الكبرى التالية والتي ستكون من المستوى الثانى (الأبناء)، وهنا نجد أن القيمة 40 هى التالية فنضعها فى المستوى الأول ونبحث عن أحد أتباعها ليحل محلها وهو إما (20 أو 17)، وبما أن القيمة 20 أكبر فننقلها إلى الخانة رقم ٢ وعليه ستكون الشجرة كالتالى :



(٣) بعد ذلك نضع القيمة الكبرى (40) فى الخلية التالية التى تم استخدامها، وعليه ستكون الشجرة كالتالى، أى تم إحلال القيمة (40) فى الخلية رقم (6) ونحتفظ بالقيمة السابقة (15) حتى نجد لها الخانة الملائمة، وذلك بتكرار الخطوة الثانية .



٤) بعد تكرار كل من الخطوات ١، ٢، ٣، سنحصل على الشجرة مفروزة كالتالى :



لاحظ أن الشجرة المفروزة بهذا الأسلوب تختلف عن شجرة العائلة بشيئين :

١) أن الخانة الأولى تحتوى على أصغر قيمة .

٢) أنها معكوسة .

والبرنامج التالى يبين كيفية استخدام هذا الأسلوب فى الفرز :

شكل (١٤-٣)

برنامج للحصول على أرقام عشوائية باستخدام دالة RND وفرزها باستخدام أسلوب HEAP

```

10 DIM A(1000)
20 GOSUB 50:REM عشوائية ارقام على ارقام للحصول
30 GOSUB 140:REM الفرز عملية اجراء
40 END
50 REM RND دالة باستخدام
60 CLS:LOCATE 10,20:PRINT "
ادخل عدد الارقام المراد الحصول عليها:
70 LOCATE 10,20:INPUT N
80 FOR I = 1 TO N
90 LET B = RND : REM
100 REM \١٠٠٠ و \١ ما بين
110 LET A(I) = INT (B*1000) للحصول على رقم صحيح يقع ما بين صف واحد
120 NEXT I
130 RETURN
140 REM HEAP نظرية باستخدام
150 LET S$=TIME$:REM وقت الفرز
160 FOR I = N TO 2 STEP -1:REM لتحديد بداية وقت الفرز
170 LET P = 1:REM اسناد القيمة \١ اموشر المصفوفة
180 FOR J = 2 TO I:REM للحصول على اقل من اعلى الى اسفل
190 REM لا اجراء عملية المقارنة بين القيم
200 IF A(J)>A(P) THEN 260
210 REM لم يتحقق الشرط فستتم عملية تغيير اماكن القيم
220 LET T = A(I)

```

```

230 LET A(I) = A(P)
240 LET A(P) = T
250 GOTO 270
260 LET P = J:REM الموشر
270 NEXT J
280 NEXT I
290 REM للحصول على وقت نهاية الفرز
300 LET F$=TIME$
310 LOCATE 13,20
320 PRINT "الوقت من heap ثم فرز " N;"من الاوقات"
330 LOCATE 15,20 :PRINT F$;"الى:"S$;"من الوقت من"
340 RETURN

```

والجدول التالي يبين الفروقات في الأوقات المستغرقة لفرز كميات من الأعداد المختلفة باستخدام كل من الأساليب الثلاثة :

تسلسل الأرقام		عشرة أرقام		خسون رقماً		مئة رقم		خمسائة رقم	
من	إلى	من	إلى	من	إلى	من	إلى	من	إلى
الفقاعي BUBBLE	00:38:07 - 00:01:09	00:10:10 - 00:10:09	01:14:40 - 01:15:04	01:10:47 - 01:12:24	00:01:09 - 00:12:24	01:01:09 - 01:12:24	00:01:09 - 00:12:24	00:01:09 - 00:12:24	00:01:09 - 00:12:24
	ثانية	ثانية	٢٤ ثانية	دقيقة ٣٧ ثانية	دقيقة ٣٦ ثانية	دقيقة ٣٧ ثانية	دقيقة ٣٦ ثانية	دقيقة ٣٧ ثانية	دقيقة ٣٦ ثانية
شل SHELL	01:17:14 - 01:17:15	01:17:15 - 01:18:28	01:18:28 - 01:18:35	01:19:41 - 01:19:41	01:19:41 - 01:19:41	01:19:41 - 01:19:41	01:19:41 - 01:19:41	01:19:41 - 01:19:41	01:19:41 - 01:19:41
	ثانية	ثانية	٧ ثوانٍ	ثانية ١٦	ثانية ١٦	ثانية ١٦	ثانية ١٦	ثانية ١٦	ثانية ١٦
الشجري HEAP	01:22:26 - 01:22:27	01:22:27 - 10:22:56	01:23:21 - 01:23:21	01:23:48 - 01:23:48	01:52:27 - 01:52:27	01:52:27 - 01:52:27	01:52:27 - 01:52:27	01:52:27 - 01:52:27	01:52:27 - 01:52:27
	ثانية	ثانية	٢٥ ثانية	دقيقة ٣٩ ثانية	دقيقة ٣٩ ثانية	دقيقة ٣٩ ثانية	دقيقة ٣٩ ثانية	دقيقة ٣٩ ثانية	دقيقة ٣٩ ثانية

ملاحظة : لقد تم تشغيل برامج الأساليب الثلاثة على جهاز آي بي إم وسعة ذاكرته الرقمية (256 K) يعتبر أسلوب الفرز (شل SHELL) من أفضل الأساليب الثلاثة بغض النظر عن عدد الأرقام (السجلات) المراد فرزها، وذلك حسب الزمن المستغرق في الفرز كما هو مبين في الجدول السابق .

البحث : في حالة البحث (Search) عن بيانات سجل معين في ملف ، سجل دراسي مثلاً ، يمكن أن يتم بطريقة تناعية Sequential أى قراءة بيانات السجل الأول ، ومقارنة بيانات الحقل المراد بالقيمة المدخلة . وفي حالة المساواة تعرض البيانات ، وإلا نستمر في قراءة بيانات سجل آخر حتى نهاية الملف فإما أن نجد السجل المراد أولاً نجده .

ولو فرضنا أن عدد السجلات الموجودة مائة سجل/ وأن بيانات السجل المراد كانت في السجل تسعة وتسعين ، فهذا يعنى أننا سنضطر للقيام بتسع وتسعين عملية مقارنة حتى نصل إلى السجل ، ولو كان عدد السجلات أكثر من ذلك فنضطر للانتظار لوقت أطول حتى نحصل على البيانات المرادة .

مثال (١٤ — ٤) : (أسلوب البحث الثنائي) :

وسنتعرض الآن لنوع آخر من طرق البحث والتي تستخدم فقط عندما تكون السجلات المراد البحث فيها مفروزة ، ويطلق عليها اسم نظرية (البحث الثنائي Binary Search) بسبب الطريقة المستعملة في عملية البحث .

فلو كان لدينا مصفوفة مكونة من ثمانية عشر سجلاً فعملية البحث باستخدام هذه النظرية تتم كالتالى :

١ — رقم السجل الأول = ١

٢ — رقم السجل النهائى = ١٨

٣ — جمع رقم السجل الأول ورقم السجل الأخير، وقسمة الناتج على ٢ والحصول على الناتج بدون كسر، لنستخدم العبارة التالية :

$$M = \text{INT} (L + H / 2)$$

حيث إن L = المتغير الذى يحتوى على رقم السجل الأول الذى سيبدأ منه البحث .
 و H = المتغير الذى يحتوى على رقم السجل النهائى الذى سينتهى إليه
 البحث .

و M = المتغير الذى يحتوى على رقم السجل الذى سنقارن فيه القيمة المدخلة بقيمة
 الحقل المرادف .

٤ — مقارنة القيمة المدخلة بقيمة الحقل ، فإذا حصلت المساواة يتم إخراج البيانات
 المرادة .

٥ — أما فى حالة عدم المساواة ، فيتم الاستفسار عما إذا كانت القيمة المدخلة أكبر من
 قيمة الحقل التى قورنت به .

٦ — ففى حالة كون القيمة أكبر ، يعنى أن البحث سيتم فى النصف السفلى من
 المصفوفة ، وعليه ستسند قيمة المتغير M مضافاً إليها واحد إلى المتغير L .

٧ — أما إذا كانت القيمة أقل ، أى أن البحث سيتم فى النصف العلوى من المصفوفة ،
 فيتم طرح واحد من قيمة المتغير M وإسناد قيمته للمتغير H .

٨ — الاستفسار عن قيمة المتغير L إذا كانت أكبر من قيمة المتغير H ، يعنى ذلك أنه قد
 تم الوصول إلى نهاية النصف الذى يجب أن يتم البحث فيه وأنه لم يتم العثور على
 ما نبحث عنه .

وفى كلتا الحالتين يتم العودة والتنفيذ من الخطوة رقم ٣ .
 فإذا كانت لدينا المصفوفة A وعدد خلاياها سبع كالتالى :

45, 32, 27, 22, 20, 15, 10

فإن عملية البحث الثنائى تتم كالتالى :

$$L = 1, H = 7$$

$$M = \text{INT} (L + H) / 2 = (1 + 7) / 2 = 4$$

$$A (M) = 22$$

		هل B أكبر من A (M) لا				
A		B	L	H	M	A (M)
1	10	20	1	7	4	22
2	15					
3	20					
4	22					
5	27					
6	32					
7	45					

هل $B = A(M)$ لا

هل A (M) أكبر من B نعم

إذن البحث سيتم في النصف العلوى

وعليه فإن: 20 1 3 2 15

هل $H = L$ لا

هل $B = A(M)$ لا

هل A (M) أكبر من B لا

إذن البحث سيتم في النصف السفلى

وعليه فإن 20 M + 1 = 3 3 3 20

هل $H = L$ لا

هل $B = A(M)$ نعم 20 = 20

وفيما يلي البرنامج الذى يعمل على البحث الثنائى :

شكل (١٤-٤)

برنامج البحث عن قيمة في مصفوفة مفرزة باستخدام أسلوب البحث الثنائي BINARY SEARCH

```

10 REM استناد القيمة 1 و 10 للمصفوفات
20 LET L = 1
30 LET H = 7
40 REM لقرائة الجيانات المفروزة و وضعها في مصفوفة
50 FOR I = 1 TO 7
60 READ A(I)
70 NEXT I
80 INPUT "ادخل الرقم المراد البحث عنه " R
100 IF L > H THEN 230
110 REM للحصول على منتصف المصفوفة
120 LET M = INT((L + H) / 2)
130 REM استفسار عما اذا تساوت القيمة المدخلة وقيمة الخانة
140 IF A(M) = R THEN 250
150 REM استفسار اذا كانت القيمة المدخلة اكبر من قيمة الخانة
160 IF A(M) < R THEN 200
170 REM طرح واحد من مكان نهائية المصفوفة
180 LET H = M - 1
190 GOTO 100
200 REM اضافة واحد امنتصف المصفوفة و جعله بداية البحث
210 LET L = M + 1
220 GOTO 100
230 PRINT "هذا الرقم " R ":" لا يوجد في المصفوفة "
240 GOTO 270
250 PRINT M ":" = "الرقم الذي ادخل "
260 DATA 10,15,20,22,27,32,45
270 END

```

و يتم تنفيذ البرنامج والحصول على النتائج كالتالى :

القيمة التى تم إدخالها 20 أدخل الرقم المراد البحث عنه

الرقم الذى أدخل = 20 موجود فى المصفوفة ومكانه = 3

تطبيق حكومى

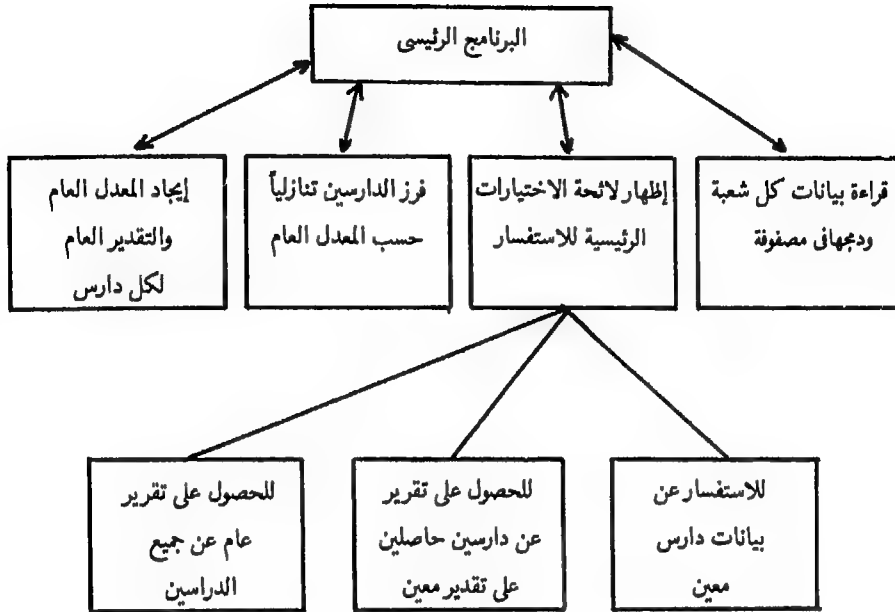
مثال (١٤ - ٥) : (الفرز والدمج والبحث) :

الهدف : شعبتان من الدارسين لبرنامج معين، أرادت إدارة القسم أن تعمل على دمج الشعبتين فى مجموعة واحدة، وإيجاد المعدل العام للامتحانين اللذين تقدم إليهما كل دارس والتقدير العام، ومن ثم فرز المجموعة تنازلياً حسب المعدل العام . وإمكانية الاستفسار عن بيانات دارس، أو الحصول على تقرير عن جميع الدارسين، أو الحصول على الدارسين الحاصلين على تقدير معين .

أولاً - خطوات الحل :

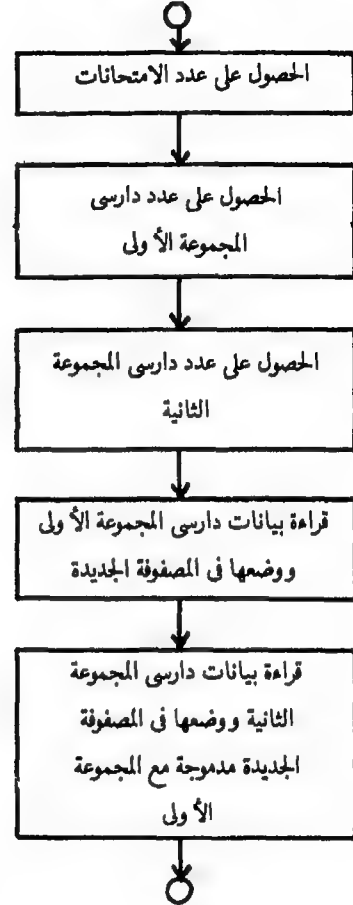
- ١ - الحصول على عدد دارسى كل من الشعبتين .
- ٢ - قراءة بيانات كل شعبة (الاسم، درجة الامتحان الأول، درجة الامتحان الثانى) . ووضعها فى مصفوفة .
- ٣ - دمج الشعبتين فى مصفوفة جديدة .
- ٤ - إيجاد المعدل العام والتقدير العام لكل دارس .
- ٥ - فرز الدارسين تنازلياً حسب المعدل العام .
- ٦ - إظهارلائحة الاختيارات الرئيسية .
- ٧ - إجراء العمليات اللازمة حسب الاختيار المدخل فى الخطوة السابقة .

ثانياً - الهيكل الهرمي للبرنامج :

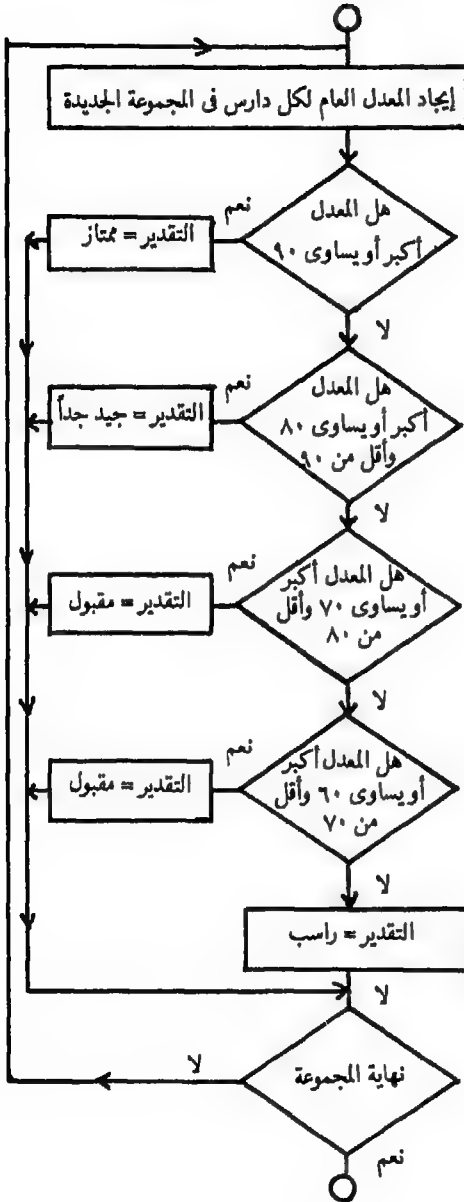


ثالثاً - نمط البرجة التركيبية :

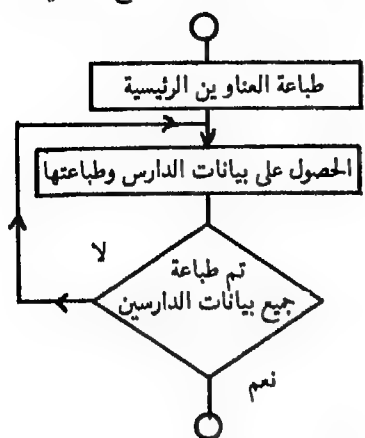
دمج المجموعتين في مجموعة واحدة



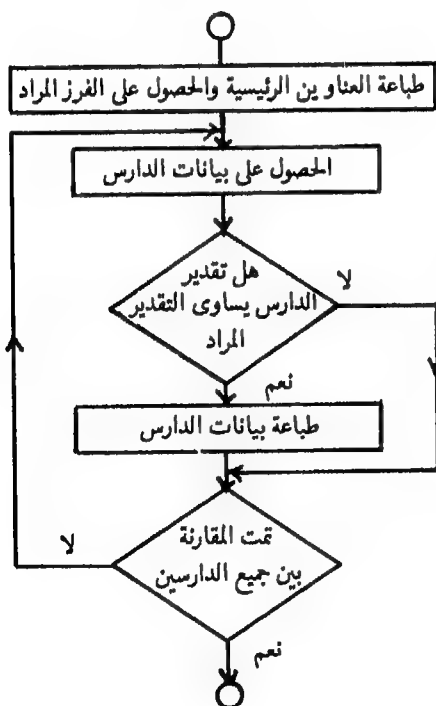
إيجاد المعدل العام والتقدير لكل دارس



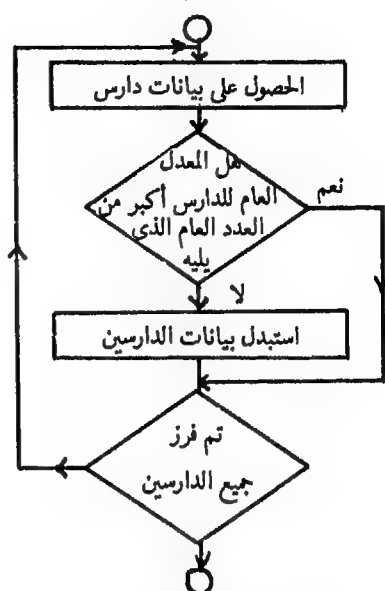
للحصول على قائمة بأسماء جميع الدارسين



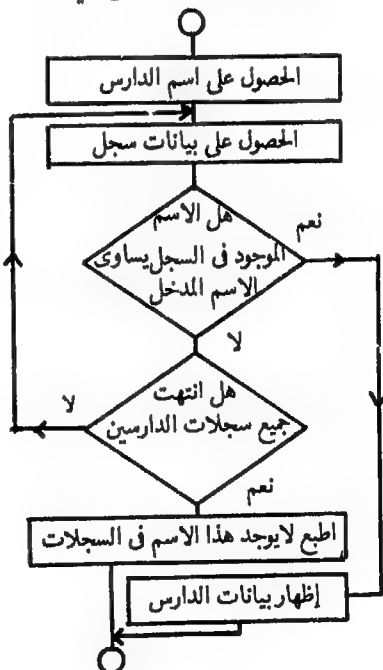
للحصول على قائمة بأسماء الدارسين
الحاصلين على تقدير معين



فرز الدارسين تنازلياً



للحصول على بيانات دارس معين



رابعاً - البرنامج كاملاً في الشكل التالي :

شكل (١٤-٥)

برنامج لدمج مجموعتين من الدارسين وإيجاد المعدل العام والتقدير لكل دارس، ومن ثم إمكانية الاستمرار

```

10 REM البرنامج الأصلي
20 DIM N$(50),N2$(50),N3$(50),B$(50),T1(50,3),T2(50,3),T3(50,3),S(50),A(50)
30 REM استخدام برنامج فرعي لفرقة المجموعة ودمجها في مجموعة واحدة
40 GOSUB 120
50 REM استخدام برنامج فرعي لايجاد المعدل العام والتقدير لكل دارس
60 GOSUB 440
70 REM استخدام برنامج فرعي لفرز المجموعة حسب المعدل العام تخانليا
80 GOSUB 620
90 REM استخدام برنامج فرعي لا جلاء العمليات الا زمه
100 GOSUB 870
110 END

120 CLS:LOCATE 8,20:PRINT " "
130 LOCATE 8,20:INPUT T "ادخل عدد الطلاب متحاننا"
140 LOCATE 10,20:PRINT T " "
150 LOCATE 10,20:INPUT N1 "ادخل عدد الدارسين في المجموعة الاولى"
160 LOCATE 12,20:PRINT N1 " "
170 LOCATE 12,20:INPUT N2 "ادخل عدد الدارسين في المجموعة الثانية"
180 FOR I=1 TO N1
190 READ N1$(I):LET N3$(I)=N1$(I)
200 FOR J=1 TO T
210 READ T1(I,J)

```

```

220 LET T3(I,J)=T1(I,J):NEXT J
230 NEXT I
240 LET I = I - 1
250 REM      B      مصفوفة في مصفوفها ووضعمها
260 REM      (      مصفوفة C )
270 FOR K=1 TO N2
280 READ N2$(K):LET N3$(I+K)=N2$(K)
290 REM
300 FOR J=1 TO T
310 READ T2(K,J):LET T3(I+K,J)=T2(K,J)
320 NEXT J
330 NEXT K
340 DATA MOHAMMAD,90,70,88
350 DATA ALI,80,90,77
360 DATA SALIH,70,60,89
370 DATA AHMED,70,80,90
380 DATA KHALID,80,90,78
390 DATA SAID,70,60,59
400 DATA MAJED,90,70,88
410 DATA NASIR,80,70,79
420 DATA WLEED,60,50,80
430 DATA OSAMA,60,70,80
440 REM
450 REM
460 REM
470 LET N=N1+N2
480 FOR I=1 TO N
490 REM
500 FOR J=1 TO T

```

للحصول على دأرسي المجموعه الثانيه ودرجاتهم ووضعمها
 ومن ثم دمجها مع مجموعه A اي استندادها لمصفوفة C
 للحصول على الدرجات و اضافتها للمصفوفه
 برنامج فرعي لحساب معدلات وتقارير دأرسي المجموعتين
 بعد دمجهم في مصفوفة C
 للحصول على عدد الدأرسين الكلي
 لحساب المعدل العام لكل دأرس

```

510 LET SD=SD+T3(J,K)
520 NEXT J
530 LET A(D)=SD/T
540 REM
550 IF A(D) >= 90 THEN G$(D)="المرتبة"
560 IF A(D) < 90 AND A(D) >= 80 THEN G$(D)="الدرجة"
570 IF A(D) < 80 AND A(D) >= 70 THEN G$(D)="الدرجة"
580 IF A(D) < 70 AND A(D) >= 60 THEN G$(D)="الدرجة"
590 IF A(D) < 60 THEN G$(D)="الدرجة"
600 NEXT I
610 FOR W=1 TO 10000:NEXT W
620 REM
630 FOR I=1 TO N-1
640 FOR J=I+1 TO N
650 REM
660 IF A(D) > A(J) THEN 850
670 REM
680 LET N4$=N3$(I)
690 LET N3$(I)=N3$(J)
700 LET N3$(J)=N4$
710 REM
720 FOR K=1 TO 3
730 LET R=T3(J,K)
740 LET T3(J,K)=T3(J,K)
750 LET T3(J,K)=R
760 NEXT K
770 REM
780 LET R2=A(I)

```

لايجاد المتقددين العلم لكل دافس

بين خالص فرعي لفرق الدافسين تان ليا حسب معدل دافس

لم يتحقق الشرط لن افستخم عملية تقييد ال ماكن

لا سبدا ال اماكن الدراجات

لا سبدا ال اماكن المعدل العلم

لا ستبدال أماكن التقدير العام

برنامج فرعى لا يظهر شاشة إلا اختيار

```
790 LET A(I)=A(J)
800 LET A(J)=R2
810 REM
820 LET G1$=G$(I)
830 LET G$(I)=G$(J)
840 LET G$(J)=G1$
850 NEXT J
860 NEXT I
870 REM
880 CLS
890 LOCATE 7,10:PRINT "=====
====="
900 LOCATE 8,10:PRINT "
910 LOCATE 9,10:PRINT "
    ج م
920 LOCATE 10,10:PRINT "-----
-----"
930 LOCATE 11,10:PRINT "
1 "
940 LOCATE 12,10:PRINT "
2 "
950 LOCATE 13,10:PRINT "
3 "
960 LOCATE 14,10:PRINT "
4 "
970 LOCATE 15,10:PRINT "=====
====="
980 LOCATE 16,10:PRINT "
990 LOCATE 16,10:INPUT O
```

للحصول على قائمة بجميع الأرقام
للحصول على قائمة بأسماء الأرقام
للحصول على بيانات الأرقام
لا نه
أدخل رقم العملية المطلوبه

```

1000 REM
1010 IF D > 0 AND D < 5 THEN 1040
1020 PRINT "
XXXXXXXXXXXXXXXXXXXX"
1030 GOTO 980
1040 ON D GOSUB 1070,1170,1330,1560
1050 IF D < 4 THEN 870
1060 RETURN
1070 REM      بدو ارسين
1080 CLS
1090 PRINT "      التحديق العام      "
1100 PRINT "
      "
1110 FOR I=1 TO N
1120 PRINT USING "\
#####.##
#####.##
#####.##
#####.##
\
"
1130 PRINT "\G$(D),A(D),T3(I,3),T3(I,2),T3(I,1),N3$(D)
1140 NEXT I
1150 FOR W=1 TO 6000: NEXT W
1160 RETURN
1170 REM      بدو ارسين
1180 CLS
1190 LOCATE 10,30:PRINT "
1200 LOCATE 10,30:INPUT E$
1210 LOCATE 2,20:PRINT "
1220 LOCATE 3,20:PRINT "
1230 REM R=
1240 LET R=0
1250 FOR I=1 TO N
1260 IF G$(D) <> E$ THEN 1290
1270 LET R=R+1

```

1280 LOCATE R+3,20:PRINT USING "&		
1290 NEXT I		
1300 LOCATE R+4,20:PRINT "		
1310 LOCATE R+5,20:PRINT R ₅ ="E\$"		
1315 FOR I = 1 TO 5000:NEXT I		
1320 RETURN		
1330 REM	برنامج فرعي للمحول على بيانات دافس معين	
1340 CLS		
1350 LOCATE 4,30:PRINT "&		
1360 LOCATE 4,15:INPUT S\$		
1370 FOR I=1 TO N		
1380 IF N3\$(I) <> S\$ THEN 1530		
1390 LOCATE 6,40:PRINT "&		
1400 LOCATE 7,40:PRINT "&		
1410 LOCATE 8,40:PRINT "&		
1420 LOCATE 9,40:PRINT "&		
1430 LOCATE 10,40:PRINT "&		
1440 LOCATE 11,40:PRINT "&		
1450 LOCATE 6,25:PRINT N3\$(I)		
1460 LOCATE 7,32:PRINT T3(I,1)		
1470 LOCATE 8,32:PRINT T3(I,2)		
1480 LOCATE 9,32:PRINT T3(I,3)		
1490 LOCATE 10,25:PRINT A(I)		
1500 LOCATE 11,25:PRINT G\$(I)		
1510 FOR W=1 TO 6000: NEXT W		
1520 GOTO 1550		
1530 NEXT I		
1540 LOCATE 6,30:PRINT "&		

```

1550 RETURN
1560 REM      العمليات
1570 CLS
1580 FOR I=30 TO 50
1590 LOCATE 7,I:PRINT "=";
1600 NEXT I
1610 FOR I=7 TO 14
1620 LOCATE I,30:PRINT ":"
1630 LOCATE I,50:PRINT ":"
1640 NEXT I
1650 FOR I=30 TO 50
1660 LOCATE 14,I:PRINT "=";
1670 NEXT I
1680 LOCATE 11,35:PRINT " مع الس ٥٥ "
1690 RETURN

```

وعند تنفيذ البرنامج نتبع الخطوات التالية :

أ – تحديد عدد الامتحانات وعدد الدارسين للمجموعتين .

- أدخل عدد الامتحانات ? 3
أدخل عدد الدارسين في المجموعة الأولى ? 4
أدخل عدد الدارسين في المجموعة الثانية ? 2

ب – ظهور لائحة الاختيارات الرئيسية :

رمز	لائحة الاختيارات الرئيسية	نوع العملية
1	للحصول على قائمة بأسماء جميع الدارسين	
2	للحصول على قائمة بأسماء الدارسين الحاصلين على تقدير معين	
3	للحصول على بيانات دارس معين	
4	لإنهاء العمليات	
?	أدخل رمز العملية المطلوبة	

جـ – وعند اختيار الرقم «1» تظهر النتائج كالتالي :

الاسم	امتحان 1	امتحان 2	امتحان 3	المعدل العام	التقدير العام
KHALID	80.0	90.0	78.0	82.7	جيد جداً
MOHAMMAD	90.0	70.0	88.0	82.7	جيد جداً
ALI	80.0	90.0	77.0	82.3	جيد جداً
AHMED	70.0	80.0	90.0	80.0	جيد جداً
SALIH	70.0	80.0	89.0	73.0	جيد
SAID	70.0	80.0	59.0	63.0	مقبول

د - وعند اختيار الرقم «3» تظهر النتائج كالتالى :

أدخل اسم الدارس المراد الاستفسار عن بياناته : ALI ?

الاسم	:	ALI
درجة الامتحان الأول	:	80
درجة الامتحان الثانى	:	90
درجة الامتحان الثالث	:	77
المعدل العام	:	82.33334
التقدير العام	:	جيد جداً

هـ - أما الاختيار رقم «2» فيتطلب حالة خاصة، حيث الصعوبة فى استخدام رمز «-» لمد الأحرف . ولكن إذا حذفنا رمز المد من التقديرات، يمكن تنفيذ البرنامج كالتالى وذلك بعد اختيار رقم «٢» من الشاشة الرئيسية .

أدخل التقدير المعين جيد جداً؟

فتظهر النتائج كالتالى :

الاسم	التقدير
KHALID	جيد جداً
MOHAMMAD	جيد جداً
ALI	جيد جداً
AHMED	جيد جداً
عدد الدارسين الحاصلين على تقدير	جيد جداً = 4

و- وللخروج من النظام ، يمكن اختيار رقم «4» ، فتظهر الشاشة التالية :



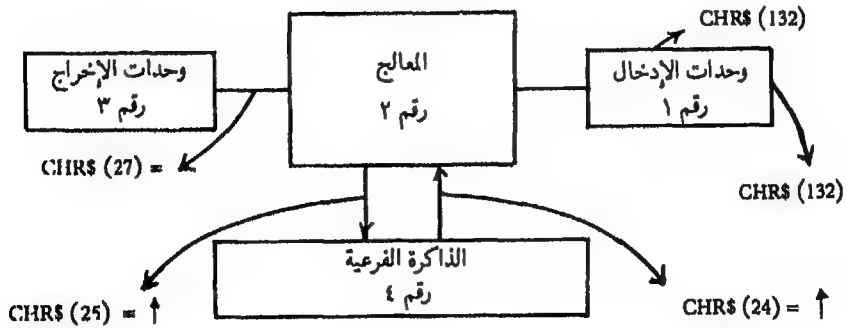
تطبيق تعليمي

مثال (١٤ - ٦) :

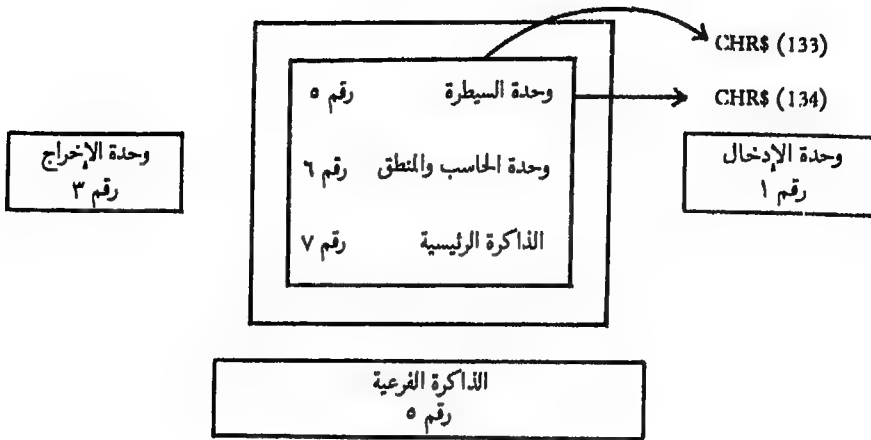
الهدف : إظهار الأجزاء الرئيسية التي يتكون منها الحاسب الآلي ومكونات كل منها ، ومن ثم إظهار كيفية انسياب البيانات والتحكم في العمليات التي يقوم بها الحاسب .

أولاً - خطوات الحل :

١ - إظهار وحدات الإدخال ، الإخراج ، المعالج والذاكرة الفرعية . كما هو مبين في الشكل التالي :

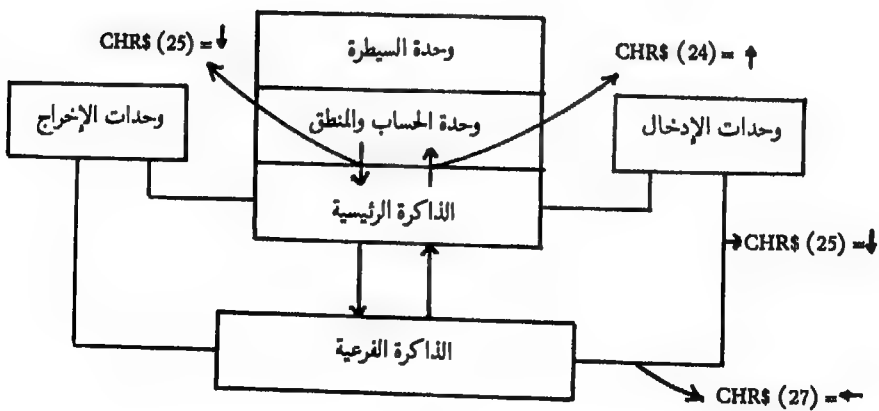


٢ - إظهار أجزاء المعالج كما هو مبين بالشكل التالي :

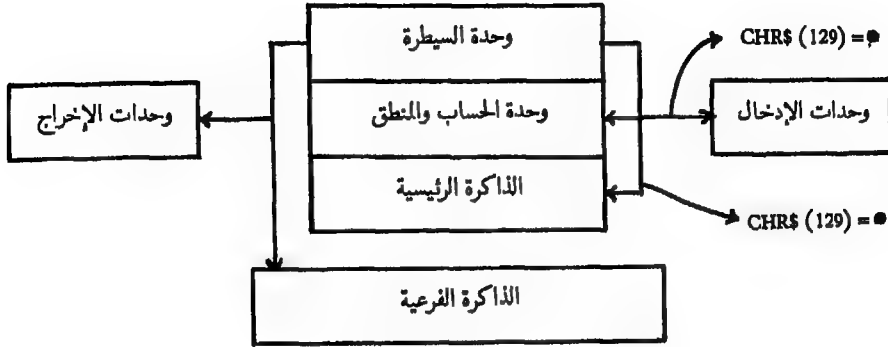


وسنشير إلى أرقام المربعات في البرنامج لتسهيل تتبع التنفيذ . كما أشير إلى نوع الرمز الذي يستخدم للخطوط كلها بواسطة الدالة CHR\$.

٣ - إظهار كيفية انسياب البيانات بين الأجزاء المختلفة :



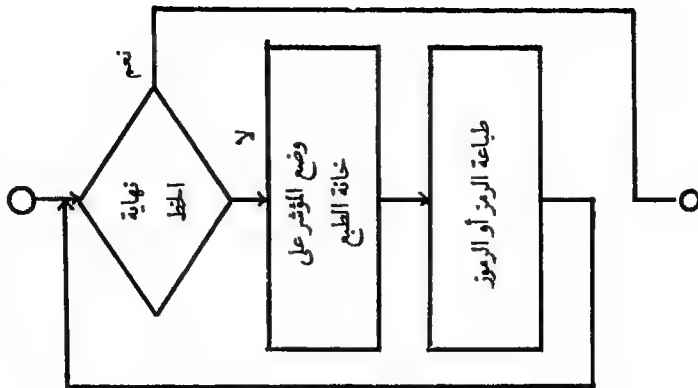
٤ - إظهار كيفية القيام بالتحكم :



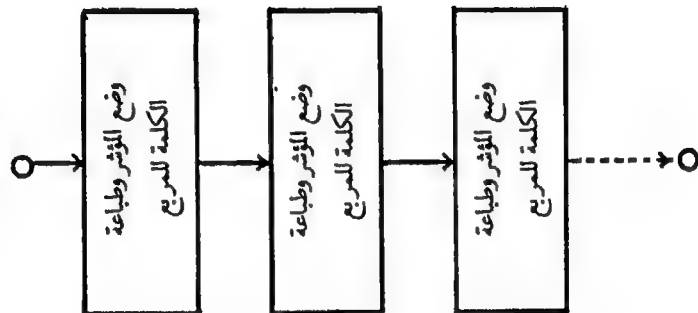
حيث سيتم استخدام الدالة CHR\$ لرسم بعض الرموز، وتعليلة COLOR للتحكم في ألوان الشاشة . وقد تم شرحهما في الفصل السادس عشر .

ثالثاً - مخطط البرمجة الترتيبية : تتبع الأجزاء في الهيكل الهرمي السابق ثلاثة أنماط رئيسية كالتالي :

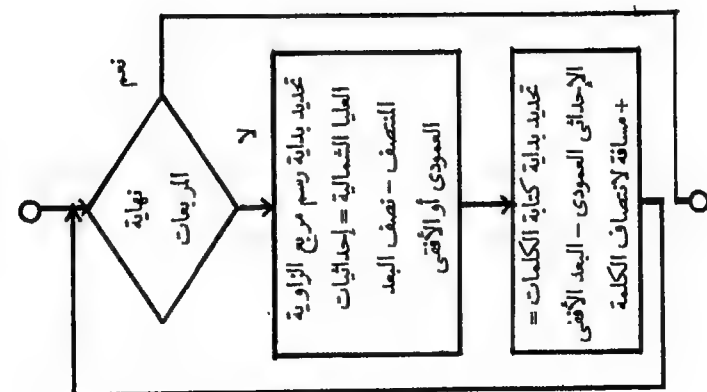
أ - الأجزاء ١، ٢، ٣، ٤، ٥، ٦، ٧، ٨، ٩، ١٠، ١١، ١٢ تتبع النمط التالي



ب - الأجزاء ٦، ٧، ٨، ٩، ١٠، ١١، ١٢ تتبع النمط التالي



ج - الأجزاء ٣، ٤، ٥، ٦، ٧، ٨، ٩، ١٠، ١١، ١٢ تتبع النمط التالي



رابعاً - البرنامج كاملاً في الشكل التالي :

شكل (١٤-١)

برنامج لإظهار الأجزاء الرئيسة للحاسب، وانساب البيانات وكيفية التحكم

```

10 KEY DIFF:REM      ألتشاشة في ألتسطح
20 CLS
30 COLOR 1,0
40 COLOR 12
50 FOR I = 1 TO 7
60 READ CNI,1,CNI,2)
70 READ DMI,1,DMI,2)
80 NEXT I
90 FOR I = 1 TO 4
100 R1 = CNI,1)- INT(DMI,1)/2)
110 R2 = CNI,2)- INT(DMI,2)/2)
120 LOCATE R1,R2
130 GOSUB 340:REM      رسم ألتخط ألتعقوي ألتمرجعات
140 GOSUB 390:REM      رسم ألتخطوط ألتعمودية
150 LOCATE R1+DMI,1),R2
160 GOSUB 340
170 FOR A=1 TO 800:NEXT A
180 NEXT I
190 GOSUB 920 REM      تعديل بداية كحابة ألتكلمات ألتأفل ألتمرجعات ألتأساسية
200 GOSUB 470 REM      تعديل بدايات ألتخطوط ألتوهلة بين ألتمرجعات
210 GOSUB 850 REM      تعديل بدايات ألتخطوط ألتوهلة بين ألتمرجعات ألتبيانات

```

```

220 GOSUB 2060 REM كتابة المعلومات داخل المبرمجيات لا سائجة
230 GOSUB 540 REM رسم المخطوطات المعلوماتية
240 LOCATE W(2,1),W(2,2)+8:PRINT " "
250 GOSUB 2130 REM تفسير ايزاء المبرمج
260 FOR P = 1 TO 500:NEXT P
270 GOSUB 2370:REM لكثافة ايزاء المبرمج
280 GOSUB 970 REM الذاكرة الرئيسية والفرعية والحساب
290 GOSUB 1220:REM رسم المخطوطات المعلوماتية
300 GOSUB 1820:REM المبرمجيات والتحكم و وحدة التحكم
310 FOR Z = 1 TO 5000:NEXT Z
320 LOCATE 25,1
330 END
340 REM المخطوطات المعلوماتية رسم
350 FOR M = 1 TO DM(1,2)
360 PRINT CHR$(132);
370 NEXT M
380 RETURN:END
390 REM المبرمجيات المعلوماتية رسم
400 FOR K = 1 TO DM(1,1)
410 LOCATE R1+K, R2
420 PRINT CHR$(132)
430 LOCATE R1+K,R2+DM(1,2)-1
440 PRINT CHR$(132)
450 NEXT K
460 RETURN:END
470 REM المبرمجيات المعلوماتية رسم
480 FOR T = 1 TO 7
490 IF T=1 OR T=2 OR T=5 THEN B(T,1)=CNT,1:B(T,2)=CNT,2)-INT(DMT,2)/2)
500 IF T=3 OR T=6 THEN 520

```



```

510 IF T=4 OR T=7 THEN B(T,2)=CNT,2:B(T,1)=CNT,1-INT(DM(T,1)/2)
520 NEXT T
530 RETURN:END
540 REM المربعات بين الخطوات رسم
550 COLOR 1,2
560 FOR Q = 1 TO 3
570 FOR S=1 TO 13
580 FOR T=1 TO 4
590 IF T=1 THEN LOCATE B(T,1),B(T,2)-S:PRINT CHR$(27)
600 IF T=2 THEN LOCATE B(T,1),B(T,2)-S:PRINT CHR$(27)
610 IF T=3 THEN 660
620 IF S > 4 THEN 660
630 IF T=4 THEN LOCATE B(T,1)-S,B(T,2)+4:PRINT CHR$(24):LOCATE B(T,1)-S+S,B(T,2)
-4:PRINT CHR$(25)
640 FOR E = 1 TO 15:NEXT E
650 SOUND 40,3
660 NEXT T
670 SOUND 32767,1
680 NEXT S
690 GOSUB 720
700 NEXT Q
710 RETURN:END
720 REM المربعات بين الخطوات مضي
730 COLOR 1,2
740 IF Q = 9 THEN 840
750 FOR C=1 TO 13
760 FOR V=1 TO 4
770 IF V=1 THEN LOCATE B(V,1),B(V,2)-C:PRINT " "
780 IF V=2 THEN LOCATE B(V,1),B(V,2)-C:PRINT " "

```

```

790 IF V=3 THEN 820
800 IF C > 4 THEN 820
810 IF V=4 THEN LOCATE B(V,1)-C,B(V,2)+4:PRINT " " :LOCATE B(V,1)-5+C,B(V,2)-4:PR
INT " "
820 NEXT V
830 NEXT C
840 RETURN:END
850 REM حساب بدائيات النقاط للمصفية
860 FOR T = 1 TO 7
870 IF T=2 OR T=5 OR T=6 THEN 900
880 IF T=1 OR T=3 THEN Z1(T,1)=CNT,1)+INT(DM(T,1)/2):Z1(T,2)=CNT,2)
890 IF T=4 OR T=7 THEN Z1(T,1)=CNT,1):Z1(T,2)=CNT,2)-INT(DM(T,2)/2)
900 NEXT T
910 RETURN:END
920 REM حساب المعلومات الخاصة بدائيات ختمية
930 FOR T = 1 TO 7
940 W(T,1)=CNT,1):W(T,2)=CNT,2)-INT(DM(T,2)/2)+1
950 NEXT T
960 RETURN:END
970 REM =====
980 COLOR ,2
990 FOR Q=1 TO 3
1000 FOR L=1 TO 5
1010 FOR Y=1 TO 4
1020 LOCATE B(4,1)-Y,B(4,2)-4:PRINT CHR$(24):LOCATE B(4,1)-5+Y,B(4,2)+4:PRINT CH
R$(25)
1030 NEXT Y
1040 REM
1050 FOR Y = 1 TO 4

```

```

1060 LOCATE B(4,1)-Y,B(4,2)-4:PRINT " "LOCATE B(4,1)-5+Y,B(4,2)+4:PRINT " "
1070 FOR H=1 TO 40:NEXT H
1080 NEXT Y
1090 NEXT L
1100 FOR L=1 TO 5
1110 FOR Y=1 TO 3
1120 LOCATE B(7,1)-Y,B(7,2)-4:PRINT CHR$(24):LOCATE B(7,1)-4+Y,B(7,2)+4:PRINT CH
R$(25)
1130 FOR H = 1 TO 30 :NEXT H
1140 NEXT Y
1150 REM
1160 FOR Y=1 TO 3
1170 LOCATE B(7,1)-Y,B(7,2)-4:PRINT " "LOCATE B(7,1)-4+Y,B(7,2)+4:PRINT " "
1180 NEXT Y
1190 NEXT L
1200 NEXT Q
1210 RETURN :END
1220 REM
1230 FOR G =1 TO 7
1240 LOCATE B(1,1),B(1,2)-G
1250 PRINT CHR$(27)
1260 NEXT G
1270 FOR H = 1 TO 5
1280 LOCATE B(1,1)-1+H,B(1,2)-7
1290 PRINT CHR$(25)
1300 NEXT H
1310 FOR P = 1 TO 6
1320 LOCATE B(1,1)+4,B(1,2)-7-P
1330 PRINT CHR$(27)

```

```

1340 NEXT P
1350 GOSUB 850
1360 FOR G=1 TO 6
1370 LOCATE Z1(7,1),Z1(7,2)-G-2:PRINT CHR$(27)
1380 NEXT G
1390 FOR G=1 TO 5
1400 LOCATE Z1(7,1)+1-G,Z1(7,2)-8:PRINT CHR$(24):NEXT G
1410 FOR G=1 TO 7
1420 LOCATE Z1(7,1)-4,Z1(7,2)-8-G:PRINT CHR$(27):NEXT G
1430 FOR G=1 TO 10
1440 LOCATE Z1(1,1)+G,Z1(1,2):PRINT CHR$(25):NEXT G
1450 FOR G=1 TO 7
1460 LOCATE Z1(1,1)+10,Z1(1,2)-G:PRINT CHR$(27):NEXT G
1470 FOR G=1 TO 6
1480 LOCATE Z1(4,1),Z1(4,2)-G:PRINT CHR$(27):NEXT G
1490 FOR G=1 TO 10
1500 LOCATE Z1(4,1)-G+1,Z1(4,2)-7:PRINT CHR$(24):NEXT G
1510 FOR P=1 TO 2000:NEXT P
1520 REM
1530 COLOR 0,2
1540 FOR G =1 TO 7
1550 LOCATE B(1,1),B(1,2)-G
1560 PRINT " "
1570 NEXT G
1580 FOR H = 1 TO 5
1590 LOCATE B(1,1)-1+H,B(1,2)-7
1600 PRINT " "
1610 NEXT H
1620 FOR P = 1 TO 6

```

```

1630 LOCATE B(1,1)+4,B(1,2)-7-P
1640 PRINT " "
1650 NEXT P
1660 FOR G=1 TO 6
1670 LOCATE Z(7,1),Z(7,2)-G-2:PRINT " "
1680 NEXT G
1690 FOR G=1 TO 5
1700 LOCATE Z(7,1)+1-G,Z(7,2)-8:PRINT " ":NEXT G
1710 FOR G=1 TO 7
1720 LOCATE Z(7,1)-4,Z(7,2)-8-G:PRINT " ":NEXT G
1730 FOR G=1 TO 10
1740 LOCATE Z(1,1)+G,Z(1,2):PRINT " ":NEXT G
1750 FOR G=1 TO 7
1760 LOCATE Z(1,1)+10,Z(1,2)-G:PRINT " ":NEXT G
1770 FOR G=1 TO 6
1780 LOCATE Z(4,1),Z(4,2)-G:PRINT " ":NEXT G
1790 FOR G=1 TO 10
1800 LOCATE Z(4,1)-G+1,Z(4,2)-7:PRINT " ":NEXT G
1810 RETURN:END
1820 REM =====
1830 COLOR 16,2
1840 FOR G=1 TO 7 STEP 2
1850 LOCATE B(5,1),B(5,2)+23+G:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
1860 FOR G=1 TO 5
1870 LOCATE B(5,1)+G-1,B(5,2)+30:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
1880 FOR G=2 TO 6 STEP 2
1890 LOCATE B(5,1)+4,B(5,2)+30+G:PRINT CHR$(129):FOR F=1 TO 50: NEXT F:NEXT G
1900 FOR G=1 TO 7 STEP 2
1910 LOCATE B(5,1),B(5,2)-G-2:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G

```

```

1920 FOR G=1 TO 5
1930 LOCATE B(5,1)+G-1,B(5,2)-9:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
1940 FOR G=2 TO 6 STEP 2
1950 LOCATE B(5,1)+4,B(5,2)-9-G:PRINT CHR$(129):FOR F=1 TO 50: NEXT F:NEXT G
1960 FOR G=1 TO 15
1970 LOCATE B(5,1)+G-1,B(5,2)-9:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
1980 FOR G=1 TO 7 STEP 2
1990 LOCATE B(5,1)+2,B(5,2)+31-G:PRINT CHR$(129):NEXT G
2000 FOR G=1 TO 8
2010 LOCATE B(5,1)+G-1,B(5,2)+30:PRINT CHR$(129):FOR F=1 TO 50:NEXT F:NEXT G
2020 FOR G=1 TO 7 STEP 2
2030 LOCATE B(5,1)+8,B(5,2)+31-G:PRINT CHR$(129):NEXT G
2040 COLOR 1,2
2050 RETURN:END
2060 FOR T = 1 TO 4
2070 IF T = 1 THEN LOCATE W(T,1),W(T,2)+1:COLOR 30,0:PRINT " مَدَّ خَلْعَ " :FOR T1=1 TO 200:NEXT T1
2080 IF T = 2 THEN LOCATE W(T,1),W(T,2)+8:COLOR 14,0: PRINT " لَمَعَلْج " :FOR T1 = 1 TO 200:NEXT T1
2090 IF T = 3 THEN LOCATE W(T,1),W(T,2)+1:COLOR 30,0:PRINT " مَفْرَجَات " :FOR T1 =1 TO 200:NEXT T1
2100 IF T = 4 THEN LOCATE W(T,1),W(T,2)+17:PRINT " عِيَّة " :FOR T1=1 TO 200:NEXT T1
2110 NEXT T
2120 RETURN:END
2130 REM مَدَّ خَلْعَ لَمَعَلْج مَفْرَجَات
2140 FOR I = 5 TO 7
2150 R1 = CN(I,1)- INT(DM(I,1)/2)
2160 R2 = CN(I,2)- INT(DM(I,2)/2)

```

```

2170 LOCATE R1,R2
2180 GOSUB 2240
2190 GOSUB 2290
2200 LOCATE R1+DMQ,1),R2
2210 GOSUB 2240
2220 FOR A=1 TO 800:NEXT A
2230 NEXT I
2240 REM      المخطط للمرجع      رقم
2250 FOR M = 1 TO DMQ,2)
2260 PRINT CHR$(133);
2270 NEXT M
2280 RETURN:END
2290 REM      المخطط للمرجع للمرجع      رقم
2300 FOR K = 1 TO DMQ,1)
2310 LOCATE R1+K, R2
2320 PRINT CHR$(134)
2330 LOCATE R1+K,R2+DMQ,2)-1
2340 PRINT CHR$(134)
2350 NEXT K
2360 RETURN:END
2370 REM      حساب الفجوة
2380 FOR T = 5 TO 7
2390 IF T=6 THEN LOCATE W(T,1),W(T,2):COLOR 14,0:PRINT " وحدة الحساب و المخطط "
OR T1=1 TO 200:NEXT T1
2400 IF T=5 THEN LOCATE W(T,1),W(T,2)+3:COLOR 14,0: PRINT " وحدة التحكم "
OR T1=1 TO 200:NEXT T1
2410 IF T=7 THEN LOCATE W(T,1),W(T,2)+1:COLOR 14,0: PRINT " وحدة الكسرة العددية "
OR T1=1 TO 200:NEXT T1

```

2420	NEXT	T	
2430	RETURN	END	
2440	DATA	8,73,6,12	(1) أحد اثنيات منتصف أبعاد مربع المدخلات
2450	DATA	8,41,12,26	(2) أحد اثنيات منتصف وأبعاد مربع المعالج
2460	DATA	8,09,6,12	(3) أحد اثنيات منتصف وأبعاد مربع المضرجات
2470	DATA	21,41,4,50	(4) أحد اثنيات منتصف وأبعاد مربع الذاكرة الفرعية
2480	DATA	04,41,2,22	(5) أحد اثنيات منتصف وأبعاد مربع وحدة التحكم
2490	DATA	06,41,2,22	(6) أحد اثنيات منتصف وأبعاد مربع وحدة الحساب والمنطق
2500	DATA	12,41,2,22	(7) أحد اثنيات منتصف وأبعاد مربع وحدة الذاكرة الرئيسية

وعند تنفيذ البرنامج، ستظهر الرسومات متتالية على الشاشة شبيهة بتلك المشروحة في مقدمة المثال.

تمارين

١ - اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ - يتسلم محل لبيع التمور طلبيات بواسطة ثلاثة طرق : الهاتف ، البريد ، وشخصي .
يبيع المحل ثلاثة أصناف من التمور . وفي نهاية كل يوم يقوم المدير بتحضير قائمة لكمية المبيعات كالتالي :

أنواع التمور:			
أسلوب الطلبية	الأول	الثاني	الثالث
الهاتف	xxx	xxx	xxx
البريد	xxx	xxx	xxx
شخصي	xxx	xxx	xxx

وإذا علم أن مصفوفة الأسعار للكيلو كالتالي :

أنواع التمور:			
أسلوب الطلبية	الأول	الثاني	الثالث
الهاتف	١٥	٢٠	٣٠
البريد	١٢	١٧	٢٥
شخصي	١٠	١٥	٢٠

وإذا كان هناك خصم على المجموع الكلى لقيمة المبيعات حسب الجدول التالى :

أنواع التمور:	الأول	الثانى	الثالث
أسلوب الطلبية	٦	٧	٩
الهاتف	٣	٥	٧
البريد	٢	٣	٤
شخصى			

فالمطلوب :

- ١ - طور برنامجاً لحساب مجموع كمية المبيعات حسب نوع التمور وحسب أسلوب الطلبية والمجموع الكلى .
- ٢ - طور برنامجاً لحساب قيمة المبيعات لكل نوع من التمور وأسلوب الطلبية والمجموع الكلى .
- ٣ - طور برنامجاً لحساب صافى الربح لكل نوع وأسلوب المجموع الكلى .
- ٤ - اعمل على إنشاء ملف ليحتوى على مجاميع بيع كل صنف من أصناف التمور الثلاثة خلال الشهر .
- ٥ - اعمل على إنشاء ملف ليحتوى على مجاميع الربح الصافى لكل صنف من أصناف التمور خلال الشهر .
- ٦ - طور برنامجاً لحساب مجموع المبيعات والأرباح لكل صنف شهرياً .

السلاسل : أساليب التعامل معها وتطبيقاتها

مقدمة عن السلاسل STRINGS

تعرضنا في الفصل السابق لكيفية إجراء المعالجات الحاسوبية على البيانات الرقمية فقط ، وكيفية إجراء معالجات الإدخال والفرز والدمج والبحث والإخراج للبيانات بنوعيتها : الرقمية ، وغير الرقمية . وتشير «السلاسل» إلى أرقام وأحرف أدخلت إلى الحاسب على شكل بيانات غير خاضعة للعمليات الحسابية ، فالبيانات «RIYADH11141 S.A.» تعتبر سلسلة مع كونها محتوية على أرقام.

وهناك عمليات خاصة يمكن إجراؤها على السلاسل ولها تعليمات ذات وظائف محددة من بينها :

- ١ — اختيار عدد معين من الأحرف بدءاً من يسار السلسلة .
 - ٢ — اختيار عدد معين من الأحرف بدءاً من يمين السلسلة .
 - ٣ — اختيار عدد معين من الأحرف من منتصف السلسلة .
 - ٤ — معرفة عدد الأحرف في سلسلة معينة .
- بالإضافة إلى ذلك ، يمكن دمج سلسلتين بواسطة علامة + .
- وسنشرح في الجزء التالي التعليمات التي تؤدي هذه الوظائف ، بالإضافة إلى تعليمات أخرى ذات علاقة بالتعامل مع السلاسل .

تعليمات السلاسل

الأيسر : إذا أردنا إدخال اسم شخص ما على ثلاث مراحل ، كالاسم الأول واسم الأب واسم العائلة ، ومن ثم طباعة الاسم في سطر واحد بالصيغة التالية :
الاسم الأول ، الحرف الأول من اسم الأب ، اسم العائلة ، فإن ذلك

يتطلب الحصول على الاسم الأول ، اسم الأب ، واسم العائلة ، كما هو موضح في العبارات التالية :

```
10 REM      قراءة الاسم الأول ، اسم الأب ، واسم العائلة
20 READ     NI$, N3$
30 DATA    «ALI», «AHMED», «ALHUSINEE»
```

أما طباعة اسم العائلة ، الحرف الأول من اسم الأب ، والاسم الأول فعند طلب الحصول على الحرف الأول فقط من اسم الأب . ولتحقيق ذلك ، يمكننا استخدام الدالة \$ LEFT كما هو مبين في الشكل التالي :

LEFT \$		الشكل العام لعبارة				
XXX	LEFT	X\$	=	LEFT \$	(X\$, X)	
↑	↑	↑	↑	↑	↑	↑
رقم السطر	تعليلة أسند	المتغير الذي ستسند له القيمة	إشارة تساوى	اليسار من	المتغير الذي يحتوى البيانات/ الكلمة	عدد الحروف المطلوبة

وعليه سنسند الحرف الأول من اسم الأب للمتغير \$ F 2 كما يلي :

```
40 LET F 2 $ = LEFT $ (N2 $,1)
```

اسم المتغير الذي يحتوى على اسم الأب

عدد الحروف المرادة من أقصى اليسار (وهنا الحرف الأول)

```
50 PRINT N3$;F2 $; N1$
```

```
60 END
```

ولو قمنا بتنفيذ هذه العبارات سنحصل على المخرجات كالتالى :

ALHUSINEE A. ALI

وفي مثالنا هذا يمكننا طباعة الأسماء دون استخدام الفاصلة أو الفاصلة المنقوطة أو غيرها من دالات التحكم فى الطباعة ، وذلك باستخدام إشارة الزائد الجبرية (+) كالتالى :

50 PRINT N3\$ + « » + F2\$ + « . » + N1\$

ولو قمنا بتنفيذ العبارات بعد التعديل على عبارة رقم ٥٠ لحصلنا على المخرجات بالشكل التالى :

ALHUSINEE A. ALI

و يطلق على هذه الطريقة اسم التوصيل CONCATINATION أى وضع هذه القيم أو قيم المتغيرات بلصق بعضها مع بعض ، ولهذا استخدمنا ما بين كل متغير وآخر وقبل إشارة الزائد علامتى التنصيص « » ووضعنا مسافتين بين العلامتين حتى يتم فصل الطباعة بعضها عن بعض .

الأيمن : إذا أردنا الحصول على حرف أو أكثر من يمين الكلمة نستخدم الدالة RIGHT\$ كما هو مبين فى الشكل التالى :

RIGHT\$		الشكل العام لعبارة الدالة				
XXX	LET	X\$	=	RIGHT\$	(X\$, X)
↑	↑	↑	↑	↑	↑	↑
رقم السطر	تعليلة أسند	المتغير الذى مستند له القيمة	إشارة يساوى	من يمين القيمة	المتغير الذى يحتوى على القيمة	عدد الحروف المرادة من اليمن للخلف

فمثلاً إذا كان المتغير A يحوى القيمة «ABDULLAH»

فإن :

— RIGHT\$ (A\$, 2) يساوى AH أى أول حرفين من يمين قيمة A\$

— LEFT\$ (A\$, 2) يساوى AB أى أول حرفين من يسار قيمة A\$

الوسط : كما أمكننا الحصول على أجزاء من كلمة من يسارها أو يمينها يمكننا الحصول على أجزاء من وسطها؛ وذلك باستخدام الدالة MID\$ كما هو مبين فى الشكل التالى :

MID\$				الشكل العام لعبارة الدالة					
XXX	LET	X\$	=	MID\$	(X\$,	X	,	X)
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
رقم	تعلية	اسم المتغير	إشارة	من منتصف	اسم المتغير	رقم الحرف	عدد		
السطر	أسند	الذي ستسند	يساوى	داخل	الذي يحتوى	المراد بدء	الحروف		
		له القيمة		المتغير	على القيمة	الإسناد من	المرادة		
						عنده			

فمثلاً : إذا كانت قيمة المتغير A\$ تساوى «GOODMORNING» فإن :

— MID\$ (A\$, 5, 7) يساوى MORNING أى الحروف السبعة من بداية العمود الخامس .

— LEFT\$ (A\$, 4) يساوى GOOD أى الحروف الأربعة من اليسار .

— RIGHT\$ (A\$, 7) يساوى MORNING أى الحروف السبعة من أقصى اليمين وللخلف .

— MID\$ (A\$, 1, 4) يساوى GOOD أى الحروف الأربعة من بداية العمود الأول .

لاحظ أنه إلى حد كبير يوجد تشابه ما بين الوظائف التي تؤديها كل من الدالات

MID\$ و LEFT\$ و RIGHT\$

فمثلاً إذا كانت قيمة المتغير A\$ تساوى

10 LET A\$ = «GOODAFTERNOON MORNING EVE»

20 PRINT LEFT\$ (A\$, 4) + « » + MID\$ (A\$, 5, 9) (١) فإن

(GOOD) + مسافة + AFTERNOON

GOOD AFTERNOON سيطلع

PRINT LEFT\$ (A\$, 4) + « » + MID\$ (A\$, 14, 7) (٢)

GOOD + مسافة + MORNING

GOOD MORNING سيطلع

40 PRINT MID\$ (A\$, 1, 4) + « » + RIGHT\$ (A\$, 3) + MID\$ (A\$, 17, 4) (٣)

GOOD + مسافة + EVE + NING

GOOD EVENING سيطلع

الطول: LEN وإذا أردنا الحصول على عدد الحروف التي تتكون منها كلمة معينة أو

اسم معين فيمكننا عمل ذلك باستخدام الدالة LEN كما هو مبين في الشكل

التالى :

LEN			الشكل العام لعبارة		
XXX	LET	X	=	LEN	(X\$)
↑	↑	↑	↑	↑	↑
رقم	تعليلة	اسم المتغير	إشارة يساوى	دالة الطول	اسم المتغير المراد
السطر	إسناد	الذى سيسند			إيجاد طوله (عدد
		له طول المتغير			الحروف التي
					يتكون منها)

فمثلاً إذا كانت لدينا البيانات التالية (خمس أسماء) وأردنا إيجاد طول كل اسم (عدد الأحرف التي يتكون منها)، نستخدم العبارات التالية :

```
10 FOR I = 1 TO 5
```

```
20 READ N$
```

```
30 LET A = LEN (N$)
```

```
40 PRINT A, «تساوى» N$ «عدد الحروف التي يتكون منها الاسم»
```

```
50 NEXT I
```

```
60 DATA «ALI», «AHMED», «HASAN», «SALIM», «TAMEEM»
```

```
70 END
```

ولو قمنا بتنفيذ هذه العبارات فسنحصل على المخرجات التالية :

3	تساوى	ALI	عدد الحروف التي يتكون منها الاسم
5	تساوى	AHMED	عدد الحروف التي يتكون منها الاسم
5	تساوى	HASAN	عدد الحروف التي يتكون منها الاسم
5	تساوى	SALIM	عدد الحروف التي يتكون منها الاسم
6	تساوى	TAMEEM	عدد الحروف التي يتكون منها الاسم

تحويل البيانات من رقمية إلى غير رقمية وبالعكس : فيما مضى تعرضنا لمعالجة البيانات غير الرقمية (خاصة الحروف)، كالحصول على أجزاء منها وتوصيلها ببعضها ببعض وما إلى ذلك، وإلى معالجة البيانات الرقمية كأرقام فقط لإجراء العمليات الحسابية عليها ومعالجتها كبيانات غير رقمية فقط في حالة وجودها بين علامتي التنصيص « » .

وستعرض الآن لكيفية تحويل هذه البيانات إلى أصلها أى (بيانات رقمية) وذلك باستخدام الدالة VAL كما هو مبين بالشكل التالى :

VAL			الشكل العام لعبارة		
XXX	LET	X	=	VAL	(X\$)
↑	↑	↑	↑	↑	↑
رقم السطر	تعليلة أسند	المتغير الذى مستند له القيمة الرقمية	إشارة تساوى	دالة القيمة الرقمية	المتغير الذى يحوى على البيانات

فمثلاً إذا كانت قيمة المتغير A\$ تساوى «8» والمتغير B\$ تساوى «7» لا يمكننا إجراء أى عملية حسابية عليها — لأنهما يحويان قيمتين تعتبران غير عدديتين ؛ لوجود هاتين القيمتين بين علامتى التنصيص . وللحصول على القيمة العددية لكل متغير نستخدم العبارات التالية :

10 LET A = VAL (A\$)

20 LET B = VAL (B\$)

وبعد تنفيذ العبارتين نجد أن المتغيرين العدديين A و B قد أسندت لهما القيمتان 8 و 7 بالتوالى . وبإمكاننا بعد ذلك إجراء أى عملية حسابية عليهما كالجمع أو الطرح ... وما إلى غير ذلك .
فمثلاً

30 PRINT A + B

سيكون ناتج الطباعة 15

أما إذا كانت لدينا بيانات عددية وأردنا تحويلها والتعامل معها كبيانات غير عددية، فنستخدم الدالة STR\$ كما هو مبين في الشكل التالي :

STR\$			الشكل العام لعبارة الدالة		
XXX	LET	X\$	=	STR\$	(X)
↑	↑	↑	↑	↑	↑
رقم السطر	تعلية أسند	المتغير الذى ستسند له القيمة بعد تحويلها	إشارة يساوى	دالة التحويل من قيمة عددية إلى غير عددية	المتغير أو القيمة المراد تحويلها

فمثلاً لو حصلنا على تاريخ اليوم بالصيغة التالية : اليوم (خانتان) ، الشهر (خانتان) ، والسنة (أربع خانات) - 19871015 - وأردنا إخراجها بالصيغة التالية ،
اليوم/ الشهر/ السنة، نستخدم العبارات التالية :

10 LET A## = 19871015

20 LET A\$ = STR\$ (A ##)

30 PRINT LEFT\$ (A\$,5) + «/» + MID\$ (A\$,6,2) + «/» + RIGHT\$ (A\$,2)

وفي حالة تحويل قيمة عددية إلى قيمة غير عددية يتم إضافة خانة إلى يسار الرقم للدلالة على إشارته (سالب/ موجب)، فإذا كان الرقم موجباً فلن تظهر الإشارة. بالإضافة إلى ذلك؛ فقد تم استخدام نوع جديد من المتغيرات (A##)، حيث يدل الرمز (##) بعد اسم المتغير على زيادة القدرة التخزينية للأرقام الصحيحة .

وهناك رموز أخرى تؤثر في تحديد القيم المخزنة في المتغيرات ويمكن الرجوع إلى ذلك في أدلة استخدام اللغة .

وكذلك فإن عدد خانات البيانات العددية المحولة إلى قيم غير عددية يكون أكبر من عدد الخانات الفعلية بوحدة .

وبعد تنفيذ هذه العبارة سنحصل على المخرجات كالتالى :

1987/10/15

أى تمت قراءة التاريخ كقيمة عددية وتم تحويلها إلى قيمة غير عددية ، ومن ثم تقسيمها إلى ثلاثة أجزاء وإخراجها بعد القيام بعملية التوصيل .

تطوير برامج باستخدام تعليمات السلسلات

مثال (١٥ - ١) :

الهدف : مدير محل معين لبيع قطع ملابس رجالية يريد معرفة عدد القطع المتوفرة عنده من الأصناف التى يبيعها ، مع العلم بأن هذه القطع مصنفة حسب التالى :

الرمز	الخاصة	الصنف :
	(خانة واحدة)	
١	— بذلة (ثلاث قطع)	
٢	— بذلة (قطعتان)	
٣	— بذلة ثلاث قطع سبورت	
٤	— بذلة قطعتان سبورت	
حسب المقاس 36,37,38,39,... 48	(ثلاث خانات)	الحجم :
	— القياس (خانتان)	
	— نوعه (خانة واحدة)	
L	طويل	
R	عادى	
S	قصير	

اللون :		(خانتان)
G	اللون الأساسي أخضر	
B	أسود	
W	أبيض	
D	طبيعة اللون — غامق	
L	فاتح	

فيصبح الرمز كالتالى :

الصنف	الحجم		اللون	
	القياس	النوع	الأساسى	الفرعى
1	3 : 7	L	G	D
2	3 : 8	R	B	L
3	:	S	W	
4	4 : 8			

أولاً — خطوات الحل :

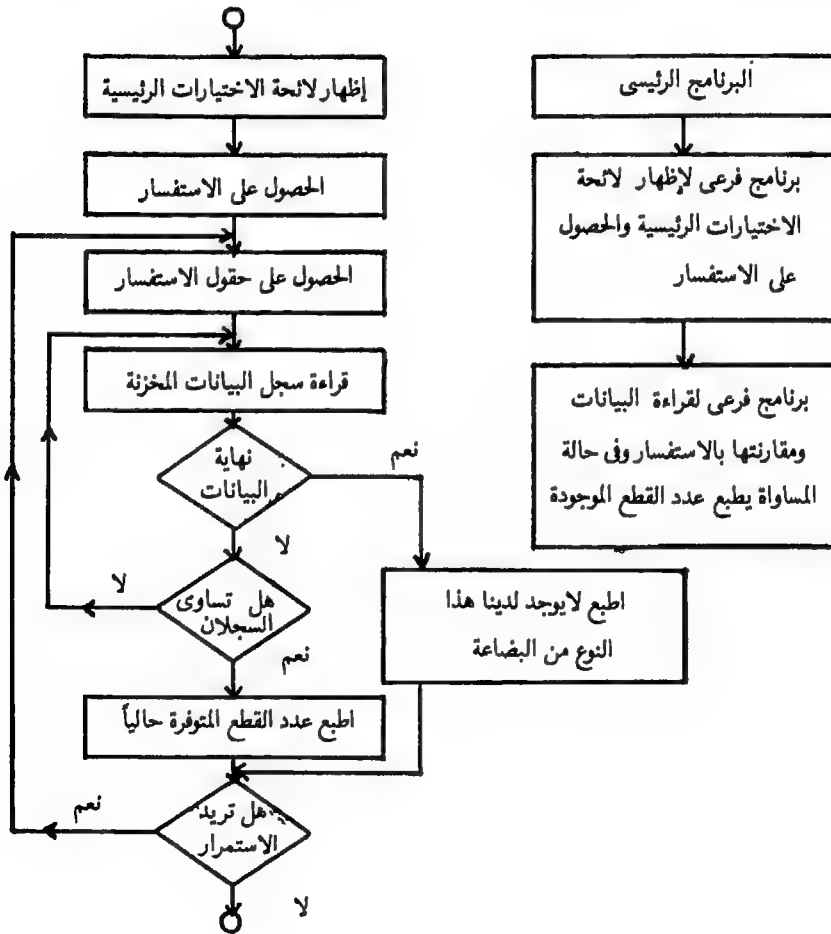
- ١ — الحصول على الاستفسار المراد .
- ٢ — معرفة رمز البدلة بالحصول على قيمة الخانة الأولى من الاستفسار .
- ٣ — معرفة الحجم المراد بالحصول على قيمتى الخانتين (٢ ، ٣) من الاستفسار .
- ٤ — معرفة نوع الحجم بالحصول على قيمة الخانة (٤) .
- ٥ — معرفة اللون المراد بالحصول على قيمة الخانة (٥) .
- ٦ — معرفة طبيعة اللون بالحصول على قيمة الخانة (٦) .
- ٧ — قراءة سجلات البيانات .

٨ - مقارنة قيم الخانات المدخلة بقيم خانات السجل (البيانات) وفي حالة المساواة نحصل على قيم الخانات (٧، ٨، ٩) من سجل البيانات لتساوى الكمية المتوفرة من القطع .

٩ - في حالة وصول نهاية البيانات وعدم حصول مساواة مع الاستفسار، يعنى ذلك عدم توفر القطعة في المخزن .

ثانياً - الهيكل الهرمى للبرنامج :

ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (١٥-١)

برنامج الاستفسار عن عدد القطع الموجودة من البذل الرجالية

```

10 REM البرنامج االس فحسي
20 GOSUB 40
30 END
40 REM برنامج فرعي لطباعة موفصفات البذل
50 CLS
60 LOCATE 3,20:PRINT "=====
70 LOCATE 4,20:PRINT " لا نكة الا فختيار االس فحسيه
80 LOCATE 5,20:PRINT "-----
90 LOCATE 6,20:PRINT " االس من
100 LOCATE 7,20:PRINT "
110 LOCATE 8,20:PRINT " 1
120 LOCATE 9,20:PRINT " 2
130 LOCATE 10,20:PRINT " 3
140 LOCATE 11,20:PRINT " 4
150 LOCATE 12,20:PRINT " 99-00
160 LOCATE 13,20:PRINT " L
170 LOCATE 14,20:PRINT " R
180 LOCATE 15,20:PRINT " S
190 LOCATE 16,20:PRINT " B
200 LOCATE 17,20:PRINT " W
210 LOCATE 18,20:PRINT " D
220 LOCATE 19,20:PRINT " L
230 LOCATE 20,20:PRINT "=====

```

```

240 LOCATE 21,20:PRINT " : لا ستفسان عنها "
250 LOCATE 21,12:INPUT A$
260 GOSUB 380:REM
270 CLS:REM المتوفره
280 LOCATE 10,20:PRINT " لا = 2 و نعم = 1 هل تريد ؟ لا ستتم ان ادخل "
290 LOCATE 10,18:INPUT A
300 IF A = 1 THEN 340
310 IF A = 2 THEN 350
320 LOCATE 20,20:PRINT "xxxxx خطئه في ؟ لا دخال xxxxxx"
330 GOTO 280
340 RESTORE : GOTO 50
350 REM
360 RETURN
370 REM
380 REM لقراءة السجل
390 READ R$
400 REM استفسان عن نهاية البيانات
410 IF R$ = " THEN 640
420 REM للممول على حقول ؟ لا ستفسان
430 LET A1$ = LEFT$(A$,1):REM نوع البطل
440 LET A2$ = MID$(A$,2,2):REM المقاس
450 LET A3$ = MID$(A$,4,1):REM نوع المقاس
460 LET A4$ = MID$(A$,5,1):REM اللون ؟ لا ساسي
470 LET A5$ = MID$(A$,6,1):REM طبعة اللون
480 REM استفسان
490 LET R1$ = LEFT$(R$,1)
500 LET R2$ = MID$(R$,2,2)
510 LET R3$ = MID$(R$,4,1)
520 LET R4$ = MID$(R$,5,1)

```

```

530 LET R5$ = MID$(R$,6,1)
540 LET R6$ = RIGHT$(R$,3)
550 REM      لا ستفسار اذا تساوت بيانات السجل مع بيانات لا ستفسار
560 IF A1$=R1$ AND A2$=R2$ AND A3$=R3$ AND A4$=R4$ AND A5$=R5$ THEN 580
570 READ R$: GOTO 410:REM      قراءه سجل آخر و الموده لل ستفسار
580 CLS: LOCATE 10,30
590 PRINT R6$;" = " عدد القاطع المرفقة "
600 FOR W=1 TO 3000: NEXT W
610 REM
620 FOR I = 1 TO 2000: NEXT I
630 GOTO 660
640 CLS: LOCATE 10,30:PRINT " يوجد هذا المصفى لا يوجد هذا المصفى
650 FOR W=1 TO 3000: NEXT W
660 RETURN
670 DATA "140LBL072"
680 DATA "238RBD090"
690 DATA "336SWL101"
700 DATA "444SBL104"
710 DATA "240RWD024"
720 DATA "350LWD073"
730 DATA "138LWL065"
740 DATA "242RBD045"
750 DATA "

```


وفي حالة تنفيذ البرنامج السابق سنحصل على التالي :

أ — الشاشة الرئيسية التي تبين مواصفات البدلة ورموزها وإدخال المواصفات التالية الموجودة في البيانات : 240 RWD .

لائحة الاختيار الرئيسية	
الرمز	الصف
1	بدلة ثلاث قطع
2	بدلة قطعتان
3	بدلة ثلاث قطع سبورت
4	بدلة قطعتان سبورت
99-00	القياس (رقمان)
L	نوع القياس — طويل
R	— عادي
S	— قصير
B	اللون الأساسي — أسود
W	— أبيض
D	طبيعة اللون — غامق
L	— فاتح

أدخل مواصفات البدلة المراد الاستفسار عنها : 240RWD ؟

ب — الإجابة على الاستفسار بالنتائج التالية :

عدد القطع المتوفرة = 024

ج — للاستمرار في الاستفسار تظهر الشاشة التالية :

هل تريد الاستمرار أدخل 1 = نعم أو 2 = لا ؟

د — الاستفسار عن مواصفات غير متوفرة، أدخل مثلاً 244RWL فتظهر الشاشة التالية :

لا يوجد هذا الصف

مثال (١٥ - ٢) :

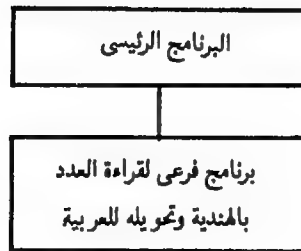
لا تحتوى لغة بيسك على القدرة على إجراء العمليات الحسابية على الأرقام الهندية (١ و ٢ و ٣ و ٩ ١) لكن لها القدرة على إجراء مثل هذه العمليات على الأرقام العربية (0, 1, 2, 9.....). لذا فى حالة وجود بيانات تحتوى على (أرقام هندية) لغرض إجراء عمليات حسابية عليها، لابد لنا من التعامل معها وكأنها قيم غير عددية، ومن ثم إيجاد المرادف لكل قيمة من الأرقام العربية، وأخيراً إجراء العمليات الحسابية على القيمة المرادفة وتحويل الناتج إلى ما يرادفه من القيم فى الأرقام الهندية.

الهدف : تحويل الأرقام الهندية إلى الأرقام العربية .

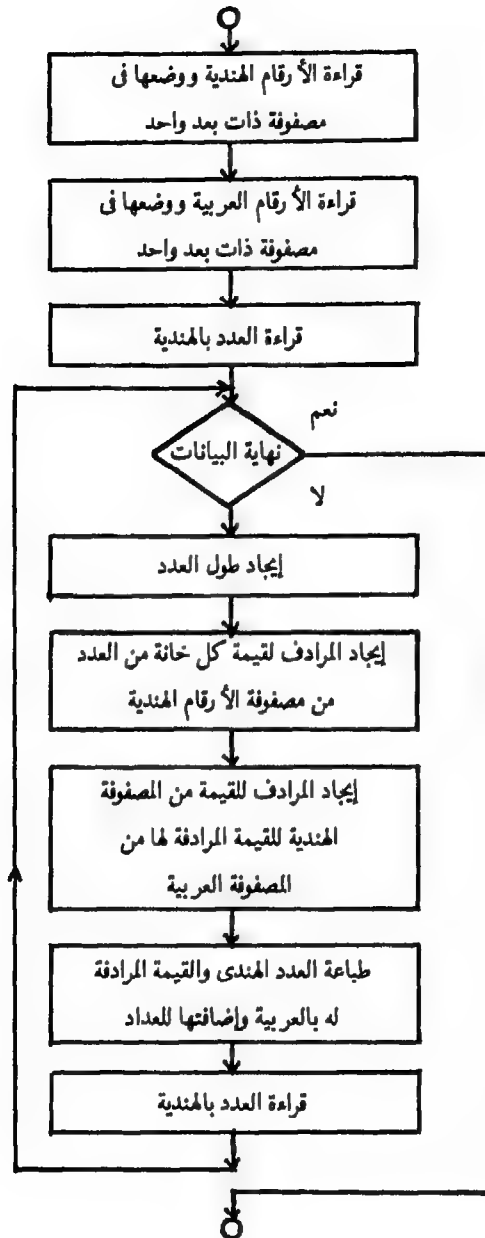
أولاً - خطوات الحل :

- ١ - الحصول على قيمة العدد .
- ٢ - إيجاد عدد خاناته (طوله) .
- ٣ - بعدد الخانات نوجد المرادف لقيمة كل خانة من الأرقام العربية .
- ٤ - توصيل الأرقام العربية ببعضها ونوجد قيمتها .
- ٥ - إخراج القيمة للعدد بالعربية والهندية .

ثانياً - الهيكل الهرمى للبرنامج :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (١٥-٢)

برنامج لتحويل مجموعة من الأعداد بالترتيب إلى ما يرادها بالمتية

```

10 REM استخدام برنامج فرعي
20 GOSUB 50
30 END
40 REM هندية
50 DIM A$(20),B$(20),C$(20),D(20)
60 FOR I=1 TO 10
70 REM لتحويل ارقام عربية الى هندية
80 READ D(I):R(I) = D(I): NEXT I
90 DATA 34,27,974,22,13,15,67,99,140,358
100 FOR I=1 TO 10
110 REM ارقام هندية ووضعتها في مصفوفة
120 READ A$(I):NEXT I
130 DATA ٩,٨,٧,٦,٥,٤,٣,٢,١,٠
140 FOR I =1 TO 10
150 IF D(I) < 100 THEN 220
160 REM للمحول على قيمة خانة للمئات
170 LET T = INT(D(I) / 100)
180 PRINT A$(T+1);
190 REM للمحول على قيمة خانة للمئات
200 LET D(I) = D(I) - (T * 100)
210 REM للمحول على قيمة خانة الآحاد
220 LET T = INT (D(I) / 10)

```

```

230 PRINT A$(T+1);
240 LET D(I) = D(I) - (T * 10)
250 T = D(I)
260 REM المخرجه و المخرجه
270 PRINT A$(T+1),R(I)
280 NEXT I
290 RETURN

```

وعند تنفيذ البرنامج، سنحصل على النتائج التالية :

٣٤	34
٢٧	27
٩٧٤	974
٢٢	22
١٣	13
١٥	15
٦٧	67
٩٩	99
١٤٠	140
٣٥٨	358
OK	

مثال (١٥ - ٣) :

التشفير CRYPTOGRAPHY هو إحدى الطرق للمحافظة على سرية البيانات والمعلومات للحيلولة دون التعرف على محتويات النص في حالة وقوعها بين أيدي غير المختصين، وتستخدم هذه الطريقة في المراسلات الحكومية وغيرها من سياسات الشركات، حيث تستخدم شفرة (CIPHER) خاصة لتحويل النص الأصلي (PLAINTEXT) إلى نص الشفرة (CIPHERTEXT) .

ويتم عمل ذلك بإحدى الطرق التالية :

- أ) طريقة الأعمدة : حيث يتم اختيار أبعاد المصفوفة المراد وضع النص بها، ومن ثم ترتيب الأعمدة المراد قراءة النص منها، ومن ثم طباعتها .
- ب) طريقة الاستبدال : حيث يتم اختيار تسلسل ترتيب الحرف (في سلسلة الحروف الأبجدية اللاتينية) الذي سيحل مكان الحرف الحالي من النص .

فمثلاً إذا كان لدينا النص الأصلي التالي NATIONAL TRANSPORTATION وأردنا كتابته بنص الشفرة بطريقة الأعمدة — فيتم ذلك حسب الخطوات التالية :

١ — تحديد نهاية النص بوضع إشارة (\$) :

فيكون النص \$ NATIONAL TRANSPORTATION

٢ — يتم تحديد أبعاد المصفوفة (الأعمدة والأسطر) المراد وضع النص الأصلي بها . بحيث يكون عدد خلايا المصفوفة مساوياً لعدد حروف النص أو أكثر .

فمثلاً لو كانت الأبعاد (6, 4) أى ستة أعمدة وأربعة سطور يصبح جدول النص كالتالى :

	1	2	3	4	5	6
1	N	A	T	I	O	N
2	A	L	T		R	A
3	N	S	P	O	R	T
4	A	T	I	O	N	\$

٣ - يتم اختيار ترتيب الأعمدة لطباعة النص، فمثلاً لو كان الترتيب المراد هو عمود (2, 1, 4, 6, 3, 5) يصبح النص كالتالي :

ALST	NANA	ITOO	NAT\$	TPI	ORRN
<u>2</u>	<u>1</u>	<u>4</u>	<u>6</u>	<u>3</u>	<u>5</u>

أما إذا أردنا كتابة النص نفسه بطريقة الاستبدال فتتبع الخطوات التالية :

١ - حذف المسافات (الفراغ) ما بين الكلمات ووضع مؤشر نهاية النص (\$) وعليه يصبح النص \$ GOODMORNING .

٢ - يتم اختيار تسلسل الحرف فمثلاً لو تم اختيار التسلسل = 3

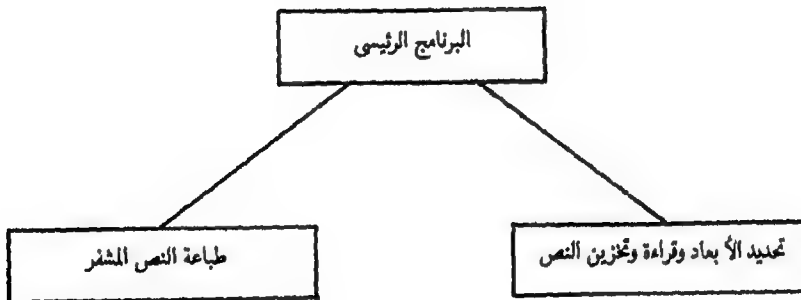
أى يتم استبدال كل حرف بالحرف الثالث الذى يليه من تسلسل الحروف الأ بجدية اللاتينية، فمثلاً لو كان الحرف من النص يساوى A يتم استبداله بالحرف D لأن ترتيب D يساوى 3 ما بعد الحرف A وهكذا . وفى حالة الوصول إلى نهاية الحروف الأ بجدية اللاتينية Z يتم العودة من بدايتها، فمثلاً لو كان الحرف من النص يساوى X يتم استبداله بالحرف A .

والبرنامج التالى سيعمل على تطبيق الطريقة الأولى :

الهدف : تشفير البيانات قبل تخزينها وإعادة فك الشفرة وطباعة النص الأصلي .

أولاً - الخطوات : كما ذكرت سابقاً.

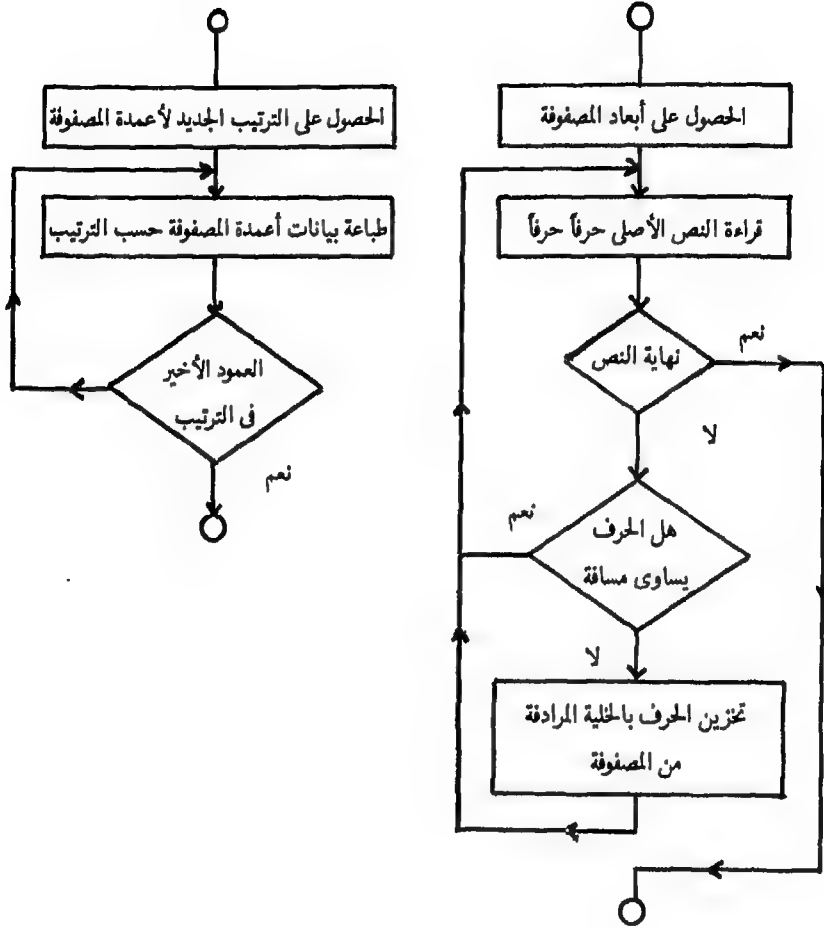
ثانياً - الهيكل الهرمى :



ثالثاً - نمط البرمجة التركيبية :

تحديد الأبعاد وقراءة النص

تشفير النص وطباعته



رابعاً - البرنامج في الشكل التالي :

شكل (١٥-٣)

برنامج لتحويل نص معين إلى نص الثفوة باستخدام طريقة الأعداد

```

10 DIM A$(20,20)
20 REM  استعداء , برنامج فرعي
30 GOSUB 70
40 REM  استعداء , برنامج فرعي
50 GOSUB 310
60 END
70 REM  برنامج فرعي لقراءة ابعاد المصفوفة و النص
80 READ R,C
90 READ R$
100 PRINT R$:REM  المدخل النص , طباعة النص
110 PRINT
120 LET S = 0
130 FOR I = 1 TO R
140 FOR J = 1 TO C
150 LET S = S + 1
160 LET B$ = MID$(R$,S,J)
170 IF B$ = "$" THEN A$(I,J) = B$
180 IF B$ = "$" THEN 220
190 LET A$(I,J) = MID$(R$,S,J)
220 NEXT J
230 NEXT I
240 FOR I = 1 TO R

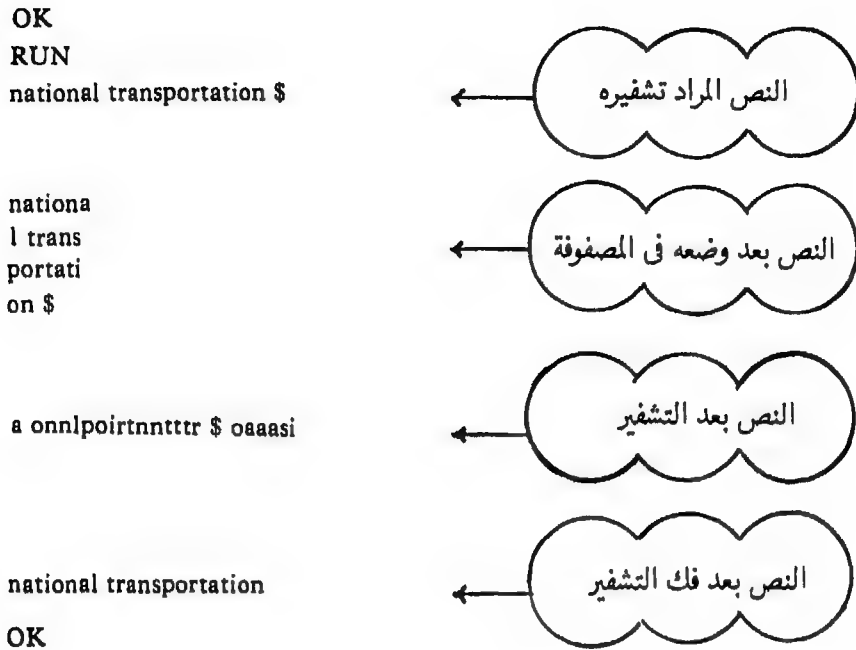
```

```

250 FOR J = 1 TO C
255 REM طباعة النقص من المصفوفة A$(I,J)
260 PRINT A$(I,J)
270 NEXT J
280 PRINT
290 NEXT I
300 RETURN
310 REM
320 FOR I = 1 TO C
330 READ P(I):NEXT I
340 FOR I = 1 TO C
350 LET J = P(I)
360 FOR K = 1 TO R
365 REM طباعة النقص المصفوف طباعة
370 PRINT A$(K,J);
380 NEXT K
390 NEXT I
400 PRINT : PRINT : PRINT
410 FOR I = 1 TO R
420 FOR J = 1 TO C
430 IF A$(I,J) = "$" THEN 460
435 REM طباعة النقص بعد إعادة تحميله
440 PRINT A$(I,J);
450 NEXT J: NEXT I
460 REM
470 RETURN
480 DATA 5,7
490 DATA "national transportation$"
500 DATA 2,1,4,6,3,5,7
510 REM

```

وفي حالة تنفيذ البرنامج في الشكل السابق سنحصل على النتائج التالية :



تعليمات أخرى للتعامل مع السلسلات

التعامل مع الأرقام الممثلة للحروف والأرقام والرموز داخل الحاسب : يتم تمثيل البيانات داخل الحاسب الآلى باستخدام النظام الثنائى (SYSTEM BINARY) وفق أنظمة مختلفة ، ومن أشهر هذه الأنظمة والتي تستخدم فى معظم الحاسبات الآلية الشخصية هو النظام الأمريكى المعيارى لتبادل المعلومات :

(AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE)

واختصاره (آسکی ASCII)

حيث يتم تمثيل كل من الحروف والرموز الخاصة والأرقام العشرية بعدد عشرى
يسهل علينا قراءته . ويمكننا الحصول على الحرف أو الرقم أو الرمز باستخدام الدالة
CHR\$ كما هو مبين بالشكل التالي :^١

CHR \$	الشكل العام لعبارة الدالة		
XXX	XXXX	CHR \$	(X)
↓	↓	↓	↓
رقم السطر	*تعليمية من تعليمات لغة بيسك	الدالة للحصول على ما يماثل قيمة الرقم العشري	الرقم العشري الذى يمثل الحرف أو الرقم أو الرمز المطلوب

• ملاحظة : تنطبق على التعليمية هنا جميع القواعد والقوانين التى تنطبق عليها فى حالة استخدامها وحدها .

(١) الجدول الذى يبين الحروف والأرقام والرموز التى يمكن الحصول عليها وما يمثّلها فى كل من النظامين (العشرى Decimal) و (السداسى عشرى Hexadecimal) يمكن إيجاده فى دليل اللغة الخاص بالحاسب المستخدم .

فمثلاً لو أردنا طباعة الحروف (الكبيرة Capital) الأ بجدية اللاتينية نستخدم
العبارات التالية :

```
10 FOR I = 65 TO 90
20 PRINT CHR $ (I);
30 NEXT I
40 END
```

ملاحظة : تم استخدام هذه العبارات مع نظام التشغيل (DOS 3.2).

وفي حالة تنفيذ هذه العبارات سنحصل على النتائج التالية :

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

كما أمكننا الحصول على الحروف والأرقام والرموز باستخدام الدالة CHR \$ وما
يمثل الحرف المراد بالنظام العشري، يمكننا الحصول على القيمة العشرية التي تماثل
حرفاً أو رقماً أو رمزاً باستخدام الدالة ASC كما هو مبين بالشكل التالي :

الشكل العام لعبارة الدالة		ASC	
(« X »)	الدالة للحصول على مايمثل الحرف	تعليمات من تعليمات لغة بيسك	رقم السطر
↓	↓	↓	↓
الحرف أو الرقم أو الرمز المراد إيجاد مايمثله	أو الرقم أو الرمز الموجود بين علامتي التنصيص	تعليمات من تعليمات لغة بيسك	رقم السطر
* ملاحظة : تنطبق على التعليمات هنا جميع القواعد والقوانين التي تنطبق عليها في حالة استخدامها وحدها .			

فمثلاً لو أردنا إيجاد مايمائل كلاً من الحروف والأرقام والرموز التالية نستخدم
العبارات التالية :

```
10  FOR I = 1 TO 10
20  READ A$
30  PRINT ASC ( « A$ » ); «      » ;
40  NEXT I
50  DATA « # », « $ », « % », « = », « W », « 7 », « ? », « g », « ( », « m »
60  END
```

وفي حالة تنفيذ هذه العبارات سنحصل على النتائج التالية :

35	36	37	61	87	55	63	103	123	109
مايمائل الرمز #									مايمائل الحرف M

الحصول على التاريخ والوقت من الحاسب : يمكننا الحصول على التاريخ والوقت من ذاكرة الحاسب الرئيسية باستخدام الدالتين DATE\$ و TIME\$ كما هو مبين في الشكل التالي :

DATE\$ أو TIME\$		الشكل العام لعبارة أى من الدالتين	
XXX	XXX	DATE\$ (1) أو	TIME\$ (2)
↓	↓	↓	↓
رقم السطر	تعليلة من تعليلات لغة البيسك	للحصول على التاريخ	للحصول على الوقت
• تنطبق على التعليلة المستخدمة جميع القواعد والقوانين التى تنطبق عليها فى حالة استخدامها وحدها .			

١ - ستكون الصيغة للتاريخ : السنة / اليوم / الشهر
 ↑ ↑ ↑
 مكون من : أربع خانات خانتين خانتين = ١٠ خانات

٢ - ستكون الصيغة للوقت : الثانية / الدقيقة / الساعة
 ↑ ↑ ↑
 مكون من : خانتين خانتين خانتين = ٨ خانات

فمثلاً لو أردنا الحصول على السنة والدقيقة نستخدم العبارات

```
10 LET Y$ = RIGHT$ (DATE$, 4)
20 LET M$ = MID$ (TIME$, 4, 2)
30 PRINT M$, «دقيقة» ; Y$, «سنة» .
40 END
```

وفي حالة تنفيذ هذه العبارات سنحصل على النتائج التالية :

سنة 1987 ودقيقة 15

كما يمكننا تغيير الوقت والتاريخ متى نشاء، وذلك بإسناد القيم المرادة لكل من الدالتين مع مراعاة الصيغة العامة لكل منهما كما يلي :

```
10 LET DATE$ = « 10/12/1986 »
20 LET TIME$ = « 08:30:10 »
```

ففى العبارة رقم 10 أصبح التاريخ فى الذاكرة الرئيسية للحاسب يساوى الثانى عشر من شهر أكتوبر لسنة ست وثمانين وتسعمائة وألف .

وفي العبارة رقم 20 أصبح الوقت في الذاكرة الرئيسية للحاسب يساوى الساعة الثامنة والنصف وعشر ثوان .

الحصول على البيانات المرحلة من لوحة الأزرار دون تحديد اسم متغير لها :

تعرضنا في السابق إلى كيفية الحصول على البيانات بطرق مختلفة سواء بواسطة تعليمة أدخل INPUT أو اقرأ READ أو الإسناد LET وعلى ضوء نوعية هذه البيانات كان يتم تحديد اسم المتغير الذى ستسند إليه (متغير عددى للبيانات العددية أو متغير غير عددى للبيانات غير العددية) .

وسنتعرض الآن للدالة INKEY\$ حيث يمكننا في الحصول على البيانات بواسطة لوحة المفاتيح حرفاً أو رقماً أو رمزاً في كل مرة نستخدمها دون إظهار هذه القيمة على شاشة الجهاز كما هو مبين في الشكل التالى :

الشكل العام لعبارة الدالة			
INKEY\$			
XXX	LET	X\$	= INKEY\$
↓	↓	↓	↓
رقم السطر	تعليمة أسند لإسناد قيمة المفتاح للمتغير	اسم المتغير الذى ستسند له قيمة المفتاح الذى سيضبط	الدالة لإشعار الحاسب باستقبال أى مفتاح من لوحة المفاتيح دون إظهار القيمة على الشاشة
	K\$		

كيفية عمل الدالة INKEY\$ متشابهة إلى حد كبير مع كيفية عمل التعليمة INPUT إلا أنهما يختلفان في :

(١) تحديد اسم المتغير ، ففى الدالة INKEY\$ تسند القيمة إلى متغير غير عددى دائماً ،

أما في تعليمة أدخل INPUT فيتم استخدام اسم متغير على ضوء نوعية البيانات المراد إدخالها .

(٢) باستخدام تعليمة أدخل INPUT يبقى الجهاز منتظراً إلى أن يتم إدخال القيمة ، أما في دالة INKEY\$ فلن ينتظر الجهاز إدخال القيمة المرادة .
وللحيلولة دون ذلك يمكننا استخدام العبارات التالية :

```
10 LET A$ = INKEY$
```

```
20 IF A$ = « » , THEN 10
```

هنا تم وضع الجهاز في حالة دوران LOOP طالما أنه لم يتم إدخال قيمة (الضغط على أى من المفاتيح) .

فمثلاً لو أردنا الاستفسار عن كلمة السر للتعامل مع برنامج معين للتأكد من السماح للمستخدم بالتعامل مع هذا البرنامج ، نعمل على إسناد كلمة السر لمتغير ومن ثم نستخدم الدالة INKEY\$ للحصول على كلمة السر ، حيث سيتم إدخالها دون إظهارها على الشاشة وهذا يقلل /يمنع من تسريبها ، ويتم ذلك باستخدام العبارات التالية :

```
10 P$ = «P302» : REM P$
```

إسناد كلمة السر للمتغير

```
20 LET L = LEN (P$) : REM
```

للحصول على طول كلمة السر

```
30 FOR I = 1 TO L
```

للحصول على كلمة السر حرفاً حرفاً

```
40 LET MID$ (E$,I,1) = INKEY$ : REM
```

إذا لم يتم الضغط على مفتاح ارجع إلى تعليمة

```
50 IF MID$ (E$,I,1) = « » THEN 40 : REM 40
```

```
60 NEXT I
```

للاستفسار عن صحة كلمة السر

```

70 IF E$ = P$ THEN 100 : REM
80 PRINT «XXX خطأ في كلمة السر لا يسمح لك بالتعامل مع البرنامج XXXX»
90 END
100 تكملة البرنامج

```

تمارين

- ١ — اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .
- ٢ — بين الأخطاء إن وجدت في كل من العبارات التالية :

10 LET A\$ = B\$ + N + N\$	أ
10 LET A\$ = LEFT («ALI» ,1)	ب
10 LET A\$ = RIGHT («ALI» ,4)	ج
10 LET A\$ = (B\$)	د
10 LET A\$ = VAL (LEFT («ALI» ,3))	هـ
10 PRINT CHAR\$ (20)	و
10 LET A\$ = MID («ALI» ,2,2)	ز

- ٣ — اكتب برنامجاً يعمل على قراءة النص التالى :

A COMPUTER IS A DATA PROCESSING MACHINE.

ومن ثم إيجاد التالى :

- أ (عدد المرات التى يتكرر بها كل حرف من الحروف الأبجدية اللاتينية .
- ب (عدد الكلمات التى يتكون منها النص .

- جـ) عدد الجمل التى يتكون منها النص .
 د) طول النص ، عدد الحروف (CHARACTERS) .
 ٤ — يتكون رمز الشقة فى مجمع كبير من الحقول التالية :
 ١ — مستوى الشقة وعدد غرف النوم لكل نوع (خانتان) .

عدد غرف النوم	النوع
٤ — ٦	ممتاز — D
٢ — ٤	متوسط — S
١ أو ٢	عادى — N

- ٢ — عدد الأبواب (خانتان) .
 ٣ — عدد الشبايك (خانتان) .
 ٤ — الموقع على بركة ماء صناعية (خانة واحدة) .
 صفر — لا ١ — نعم .
 ٥ — نوع المستأجر (خانة واحدة) . غير مؤجر صفر
 شخصى لفرد ١
 شخصى لعائلة ٢
 شركة ٣

- ٦ — رقم الشقة (ثلاث خانات)
 طور برنامجاً لحساب التالى :
 ١ — عدد الشقق غير المؤجرة لكل نوع .
 ٢ — عدد الشقق المؤجرة حسب نوع المستأجر .
 ٣ — عدد الأبواب والشبايك فى المجمع لأغراض الصيانة .
 ٤ — عدد الشقق غير المؤجرة والموجودة على بركة الماء .

من الأمثلة على البيانات المدخلة :

D54600112

S33512433

N13401315

٥ — يتكون رمز الحجرة في المستشفى المركزى من الحقول المرمزة التالية :

(١) نوع الحجرة (خانة واحدة) P خاص

S شبه خاص G عام

(٢) إذا كان غير خاص ، فعدد الأسرة في الحجرة (خانة واحدة) . شبه الخاص يكون العدد ٢ أو ٤ و العام ٤ — ٧ أسرة .

(٣) هل الحجرة فيها هاتف أم لا (خانة واحدة) T مع هاتف N دون هاتف .

(٤) هل الحجرة فيها تلفاز (خانة واحدة) V مع تلفاز N دون تلفاز .

(٥) هل الحجرة فيها تكييف خاص أو تابعة C مركزى للتكييف المركزى D محلى

(٦) رقم الطابق (خانة واحدة) هناك ثلاثة طوابق في المستشفى .

(٧) رقم الحجرة (خانتان) .

(٨) مؤشر للدلالة عن مدى شعور الأسرة في الحجرة 0 دون شواغر 1 - 7 عدد الشواغر .

المطلوب :

١ — أوجد عدد الأسرة الموجودة في الحجرات الخاصة وشبه الخاصة والعامة .

٢ — أوجد عدد الأسرة الشاغرة في كل نوع من أنواع الحجر .

٣ - أوجد عدد الأسرة فى الحجرات (شبه الخاصة أو العامة) وفى كل منها هاتف .
٤ - أوجد عدد الأسرة فى الحجرات التى فى الطابق الأول والتى فى كل منها هاتف ،
وخاصة .

٥ - أوجد عدد الأسرة فى الحجرات التى فى كل منها هاتف وتلفاز .
٦ - أوجد عدد الأسرة فى الحجرات الخاصة والتابعة للتكييف المركزى .
٧ - أوجد عدد الأسرة فى الحجرات التى فى كل منها (تلفاز أو هاتف) و (تقع فى
الطابق الأول أو الثانى) .

من الأمثلة على المدخلات :

P1TND2140

C4NNC3153

S2TVC2162

الرسومات البيانية

مقدمة عن الرسومات البيانية

في الفصول السابقة ، تم استخدام الأرقام والجداول لعرض المعلومات إما على الشاشة أو على الورق ، ومع أن استخدام الرسومات البيانية لعرض المعلومات قديم عهد في الإحصاء ، إلا أنه حديث عهد في الحاسب .

وتمتاز لغة البيسك عن غيرها من اللغات بتوفر تعليمات برمجة تساعد في عمل رسومات على الشاشة . ومن أنواع هذه الرسومات التي يمكن رسمها في لغة البيسك الدوائر المقسمة والخطوط والبيانات والأعمدة .

فوائد استخدام الرسومات البيانية

ومن فوائد استخدام الرسومات لإظهار نتائج المعالجات بدلاً من القيم والجداول :

١ - اختصار الوقت في الاطلاع على نتائج التقارير وخاصة للمديرين الذين ليس لديهم الوقت الكافي لفعل ذلك ، فالرسومات البيانية يسهل الاطلاع عليها بطريقة أسرع وأجدى .

٢ - سهولة استنتاج الحالات الغريبة والخارجة عن النمط العام للبيانات ، كزيادة مبيعات شهر رجب عن غيره من الشهور .

٣ - سهولة المقارنة بين البيانات التابعة لظواهر مختلفة ، والتي تحتوى في معظم الحالات على بيانات كثيرة ، كالمقارنة بين مبيعات مناطق مختلفة تابعة لشركات متعددة .

- سهولة استنباط نمط عام (اتجاه) للبيانات ، مثل ملاحظة الازدياد المطرد للمبيعات خلال السنوات الخمس الماضية .

٥ - إمكانية استخدام الألوان لتسهيل التوصل إلى النتائج السابقة .




وقبل تطوير برامج للرسومات البيانية ، سنتعرض لأوضاع شاشة العرض وكيفية التحكم فيها .

أوضاع الشاشة

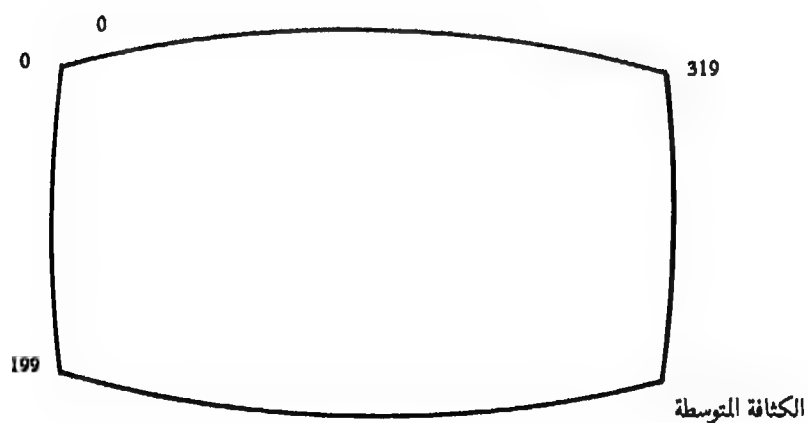
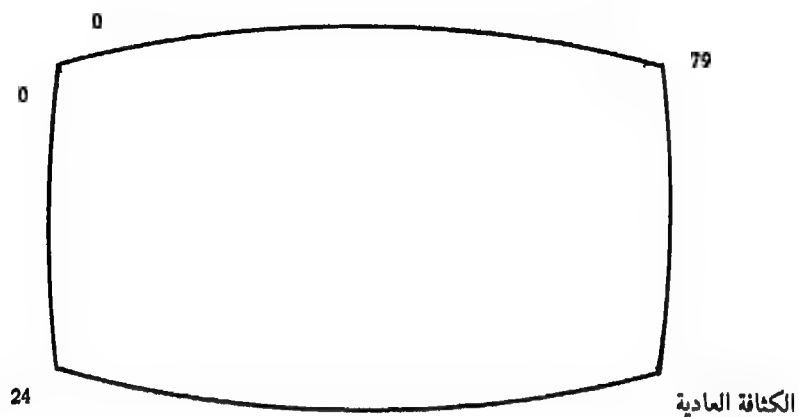
للشاشة ثلاثة أوضاع تحدد عدد الأسطر والأعمدة فيها : كلما زاد عدد الأسطر والأعمدة ارتفعت كثافة الشاشة وقلت إمكانية استخدام الألوان . ولتحقيق الكثافة فوق العادية على الحاسبات الشخصية ، لابد من وجود شاشة ملونة خاصة بالرسومات ، أما الشاشات العادية ، ففيها الكثافة العادية فقط . و يوضح الجدول التالى أوضاع الشاشة :

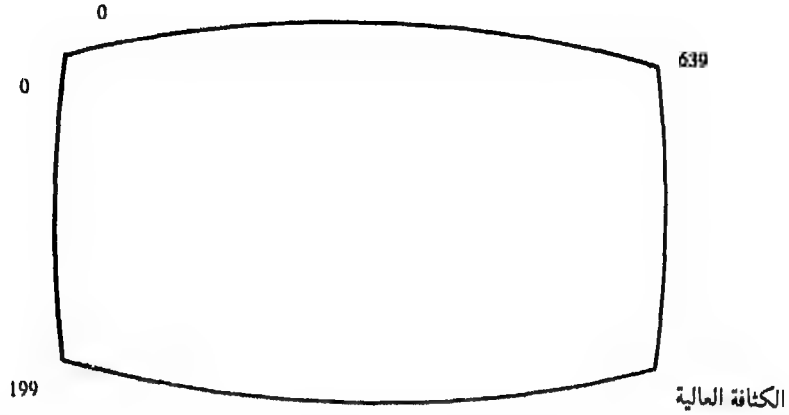
كثافة الشاشة		SCREEN RESOLUTION	
نوع الشاشة	عادية TEXT	متوسطة MEDIUM	عالية HIGH
	٨٠ عموداً	٣٢٠ عموداً	٦٤٠ عموداً
	٢٥ سطرأ	٢٠٠ سطر	٢٠٠ سطر
عادية	١٦ لوناً	غير متاح	غير متاح
ملونة خاصة بالرسومات	١٦ لوناً	٤ ألوان	لونان أبيض وأسود

وتحدد الكثافة بمقدار حجم الخلية المستخدمة للرسم البيانى والناجمة عن تقاطع السطر والعمود و يوضح ذلك الشكل التالى :

الكثافة العادية	الكثافة المتوسطة	الكثافة العالية	مجموع الخلية الواحدة النسبى
٢٥ × ٨٠	٢٠٠ × ٣٢٠	٢٠٠ × ٦٤٠	
			

ويمكن تصوير حجم هذه الخلايا على الشاشة ذات البعدين الثابتين كما هو موضح في الأشكال التالية :





لاحظ أن الكثافة العادية تستخدم بشكل رئيسي في حالة كتابة النصوص والبرامج ، وهناك الكثافة تحت العادية (٤٠ عموداً و ٢٥ سطراً) ، والتي تستخدم أحياناً لطباعة الحروف والكلمات بالحجم الكبير ، وذلك لتسهيل الرؤية والقراءة والمتابعة للعمليات المصاحبة لتنفيذ البرنامج .

التحكم في أوضاع الشاشة

ولوضع الشاشة في أى من الحالات الثلاث ، واعتماداً على توفر الشاشة الملونة من عدمه تستخدم تعليمة «الشاشة SCREEN» ، كما هو مبين في الشكل التالى :

الشكل العام لتعليمة SCREEN		
XXX	SCREEN	N
↓	↓	↓
رقم السطر	التعليمة	متغير

• ملاحظة : يأخذ المتغير N أحد الأرقام التالية :

- 0 : الكثافة العادية ٨٠ عموداً و ٢٥ سطراً
- 1 : الكثافة المتوسطة ٣٢٠ عموداً و ٢٠ سطراً
- 2 : الكثافة العالية ٦٤٠ عموداً و ٢٠٠ سطر

وهناك توصيفات إضافية للشاشة يمكن الحصول عليها من دليل اللغة الخاص بكل جهاز ، وأوردنا هنا ما يهمنا فقط في هذا الفصل .

تطوير برامج على الرسوم البيانية

سنقدم في هذا القسم عدة أمثلة عن كيفية استخدام تعليمات الرسوم البرمجية لعمل رسومات تمثل نتائج المعالجات في البرنامج .

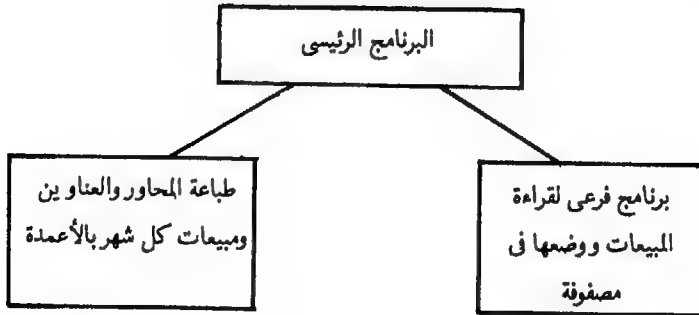
مثال : (١٦ - ١) : (الأعمدة) .

الهدف : إظهار مبيعات شركة معينة خلال الاثنى عشر شهراً للسنة الماضية .

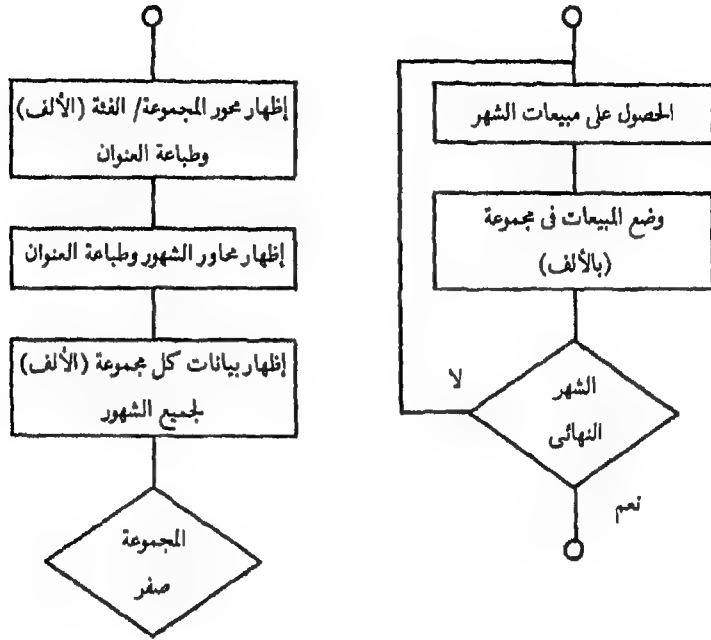
أولاً - خطوات الحل :

- ١ - تحديد محاور الشهور والمبيعات وطباعتها .
- ٢ - الحصول على مبيعات كل شهر ووضعها في مصفوفة وفئات (بالألف ريال) .
- ٣ - تمثيل المبيعات في أعمدة وطباعتها .

ثانياً - الهيكل الهرمي :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (١٦-١)

برنامج لإظهار البيانات الشهرية على شكل أعمدة

```

10 DIM A(20),B(20): KEY OFF: CLS
20 GOSUB 50: REM استعمال برنامج فرعي لقراءة المبيعات لكل شهر
30 GOSUB 120: REM استعمال برنامج فرعي لإظهار المبيعات لكل شهر
40 END
50 REM برنامج فرعي لقراءة مبيعات كل شهر و وضعها في مصفوفة
60 FOR I=1 TO 12
70 READ A(I)
80 REM لوضع مبيعات كل شهر في مجموعة بيانات لف
90 LET B(I) = A(I) / 1000
100 NEXT I
110 RETURN
120 REM ( بيانات عمدة )
130 REM لطباعة المبيعات و محور المبيعات
140 LOCATE 1,1: PRINT "المبيعات بالآلاف"
150 FOR I = 1 TO 12 STEP 1
160 LOCATE I,10: PRINT CHR$(219)
170 NEXT I
180 REM لطباعة المبيعات و محور
190 FOR I = 1 TO 12
200 LOCATE 21,I: PRINT CHR$(176);

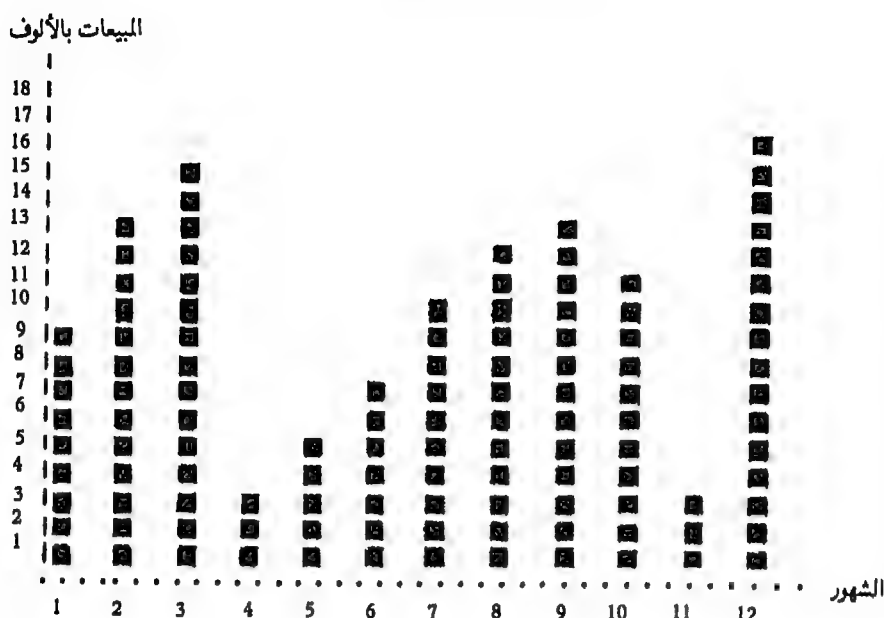
```

```

210 NEXT I
220 FOR I = 1 TO 12
230 LOCATE 22,10 + (I - 1) * 4: PRINT USING "##";I;
240 NEXT I
250 LOCATE 22,10 + (I - 1) * 4: PRINT "جش %f"
260 REM جش 5 مبعلا ج ط %
270 FOR I = 1 TO 12
280 FOR J = 20 TO 30 - 2 * BD STEP -1
285 LET Y = 11 + ( I - 1 ) * 4
290 LOCATE J,Y
300 PRINT CHR$(32);
310 NEXT J
320 NEXT I
330 RETURN
340 DATA 09000,11000,12000,6000,07000,8300
350 DATA 9500,10600,11300,10000,6400,12700

```

وفي حالة تنفيذ البرنامج في الشكل السابق، سنحصل على النتائج التالية :



كما نلاحظ من نتائج المعالجة (في كلا المثالين ١٠ - ١، ١٠ - ٢) أنه من السهل ملاحظة وإيجاد الشهر الذي تمت فيه أعلى قيمة مبيعات، سواء كان ذلك للشركة أو لأى من المندوبين الثلاثة، وأن الوقت المستغرق لإيجاد أفضل شهر أو أسوأ شهر أقل بكثير من الوقت الذى سيستغرق في حالة كون نتائج المعالجة (المخرجات) رقمية .

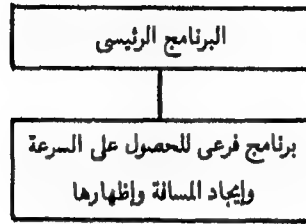
مثال (١٦ - ٢) : (النقاط المبعثرة Scatterdiagram)

الهدف : إظهار طول المسافة التى يحتاج إليها السائق لإيقاف سيارته على حسب السرعة التى يسير بها والتى تتراوح ما بين ٢٠ إلى ٨٠ كيلومتراً في الساعة .

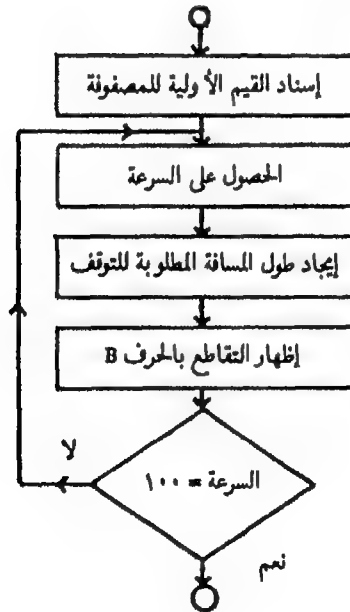
أولاً - خطوات الحل :

- ١ - الحصول على السرعة .
- ٢ - إيجاد طول المسافة .
- ٣ - إظهار تقاطع السرعة والمسافة بالـ (٥) .

ثانياً - الهيكل الهرمي :



ثالثاً - نمط البرمجة التركيبية :



رابعاً - البرنامج في الشكل التالي :

شكل (١٦-٢)

برنامج لإظهار الساعات اللازمة لإيقاف السيارة حسب سرعتها

```

10 CLS
20 SCREEN 2,1,0
30 DIM T(20),S(12),P(12),A(12)
40 GOSUB 70 :REM : إستخدماء برنامج فرعي لقراءة البيانات
50 GOSUB 210 :REM : إستخدماء برنامج فرعي لطباعة المخرجات
60 END
70 REM : إستخدماء برنامج فرعي لإيجاد النسبة المئوية لمبيعات كل الشهر
: برنامج فرعي
80 LET T=0
90 FOR I = 1 TO 12
100 READ S(I)
110 LET T = T + S(I) :REM : إضافة مبيعات الشهر لمجموع المبيعات
120 NEXT I
130 REM : لإيجاد النسبة المئوية لمبيعات كل شهر
140 FOR I = 1 TO 12
150 LET P(I) = (S(I) / T)
160 NEXT I
170 REM : البيانات
180 DATA 6000,10000,13000,7500,10500,14000
190 DATA 9300,3500,10000,5000,11000,14000

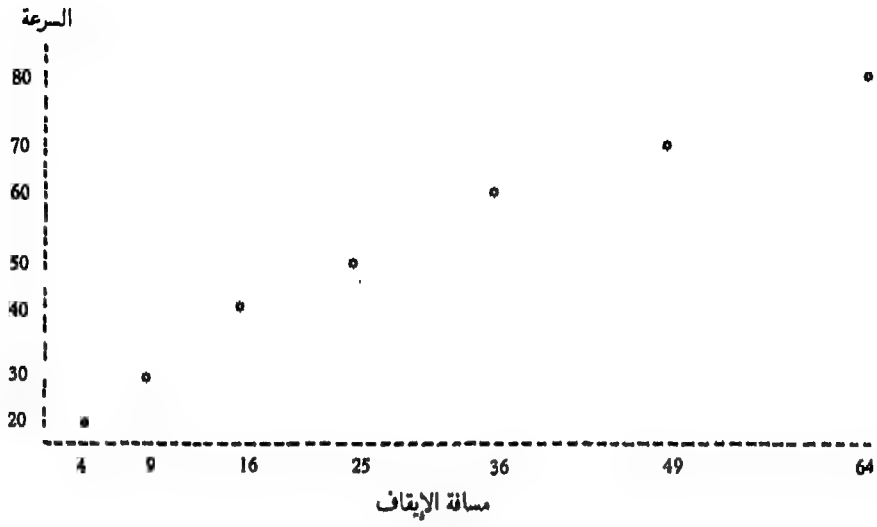
```

```

200 RETURN
210 REM      بالرسوم الدائرية
220 FOR I=0 TO 12 : REM      شهر
230 LET A(I) = 0
240 NEXT I
250 REM      إيجاد الزاوية لمبيعات كل شهر
260 FOR I = 1 TO 12
270 LET D = P(I) * (2 * 3.141)
280 LET A(I) = A(I - 1) + D
290 CIRCLE (400,100),80,1,-A(I),-A(I-1)
300 NEXT I
310 REM      لطباعة النسبة المئوية لمبيعات كل شهر
320 LOCATE 1,1: PRINT " النسبة المئوية لمبيعاته "
330 LOCATE 2,1: PRINT "===== "
340 FOR I=1 TO 12
350 PRINT USING "###" # # " ,P(I), I
360 NEXT I
370 RETURN

```

وفي حالة تنفيذ البرنامج، ستظهر النتائج التالية :



مقدمة عن استخدام الألوان في الرسومات

للحصول على نتائج المعالجة بالألوان يشترط أن يكون للجهاز إمكانية عمل ذلك بوجود شاشة ملونة .

ففى جهاز IBM PC /XT يمكننا الحصول على الألوان التالية فى حالة الكثافة العادية :^١

رمز اللون	نوعه	رمز اللون	نوعه
0	أسود BLACK	8	رمادى GRAY
1	أزرق BLUE	9	أزرق فاتح LIGHT BLUE
2	أخضر GREEN	10	أخضر فاتح LIGHT GREEN
3	أزرق داكن CYAN	11	أزرق فاتح LIGHT CYAN
4	أحمر RED	12	أحمر فاتح LIGHT RED
5	بنفسجى MAGNETA	13	بنفسجى فاتح LIGHT MAGNETA
6	بنى BROWN	14	أصفر YELLOW
7	أبيض WHITE	15	أبيض ناصع HIGH - INTENSITY WHITE

وأما فى حالة اختيار الكثافة المتوسطة فيكون اختيار الألوان كالتالى :

(0 — أسود، 1 — أزرق، 2 — بنفسجى، 3 — أبيض) أما الكثافة العالية فيكون اختيار اللونين (0 — أسود و 1 — أبيض) .

ولاختيار اللون نستخدم تعليمة لون COLOR كما هو مبين بالشكل التالى :

(1) IBM Personal Computer Hardware Reference Library BASIC, p. 39

الشكل العام لعبارة تعليمة لون				
COLOR				
XXX	COLOR	N, 1	N, 2	N 3
↓	↓	↓	↓	↓
رقم السطر	لون	لون نتائج المعالجة التي ستظهر بها	لون الشاشة	لون حافات الشاشة

- ١ - و يطلق عليه FOREGROUND ويجب أن يكون الرقم ما بين (صفر - ١٥) .
- ٢ - و يطلق عليه BACKGROUND ويجب أن يكون الرقم ما بين (صفر - ٧) .
- ٣ - و يطلق عليه BORDER ويجب أن يكون الرقم ما بين (صفر - ١٥) .

تطوير برنامج باستخدام تعليمة لون COLOR

يمكن تعديل البرنامج السابق وعرض الرسومات بالألوان كالتالى :

إذا أردنا عرض البيانات بالألوان، استخدمنا العبارة التالية فى شكل (١٦ - ٢) :

25 COLOR 0, 4, 2

فنجده أن المخرجات ستظهر باللون الأسود (●) ولون الشاشة سيكون أحمر (٤) ولون حافات الشاشة سيكون أخضر (٢) .

وهنا سيستمر عرض جميع المخرجات بهذه الألوان إلى أن يتم استخدام تعليمة لون COLOR مرة أخرى فى البرنامج .

فلعرض المخرجات بالألوان للمندوبين الثلاثة (العملية رمز ٥) فى شكل (١٦ -

٢) نستخدم العبارة التالية :

1215 COLOR J- 1, 14, 12

وفي حالة تنفيذ البرنامج ستظهر مبيعات المندوب الأول باللون الأسود، ومبيعات المندوب الثاني باللون الأزرق، ومبيعات المندوب الثالث باللون الأخضر، وسيكون لون الشاشة في جميع الحالات أصفر (١٤)، ولون الحافات أحمر فاتحاً (١٢)*.

تعليمات رسم الأشكال الهندسية مسبقة التحديد

للحصول على الأشكال الهندسية المختلفة كالدوائر والمربعات والمثلثات والخطوط نستخدم التعليمات التالية في حالتى الكثافة المتوسطة والعالية :

الخط المستقيم : إذا أردنا رسم خط نستخدم تعليمة خط LINE كما هو مبين بالشكل التالى :

الشكل العام لعبارة تعليمة خط		LINE	
(N1 , N2)	(N3 , N4)	LINE	XXX
إحداثيات نقطة الانطلاق	إحداثيات نقطة النهاية	تعليمة خط	رقم السطر
N1 العرض على الشاشة (X) N2 الطول على الشاشة (Y)	N3 — العرض على الشاشة (X) N4 — الطول على الشاشة (Y)		

إن أى نقطة على الشاشة تمثل بقيمتين و يطلق عليهما «إحداثيات»، الأولى تحدد العرض (X) أى العمود Column، أما الثانية فتحدد الطول (Y) أى السطر ROW وهذه الإحداثيات تمثل ما يطلق عليه اسم خلية PIXEL .

* ملاحظة : للحصول على مزيد من توصيفات تعليمة لون COLOR يمكن العودة إلى دليل لغة بيسك الخاص بالجهاز المستخدم .

فبالعبارتين التاليتين :

10 SCREEN 1

20 LINE (12,100) - (100,150)

نعمل على رسم خط مستقيم يبدأ من النقطة التى إحداثياتها (12,100) و ينتهى فى النقطة التى إحداثياتها (100, 150) ، أى أن المؤشر انطلق من مركز الإحداثيات الأولى إلى اليمين بثمان وثمانين PIXEL وإلى الأسفل بخمسين PIXEL كما هو مبين فيما يلى :



ويمكننا رسم خط آخر مرتبط بالخط الذى انتهينا من رسمه باستخدام تعليمة خطوة STEP كما هو مبين بالشكل التالى :

STEP		الشكل العام لعبارة تعليمة خطوة	
XXX	LINE -	STEP	(N1, N2)
رقم السطر	تعليمة خط	تعليمة خطوة	إحداثيات النقطة المراد تكملة رسم الخط لها

فإذا أردنا تكملة رسم الخط الذي انتهت إحداثياته بالقيمتين (100,150) بخمسين وحدة إلى اليمين وعشرين وحدة إلى أعلى ، نستخدم العبارة التالية :

30 LINE - STEP (50,20)

فإذا أردنا أن يكون لون الخط الجديد أزرق فستصبح العبارة كالتالي :

30 LINE - STEP (50, - 20), 1

وعليه سيصبح الخط الجديد كالتالي :



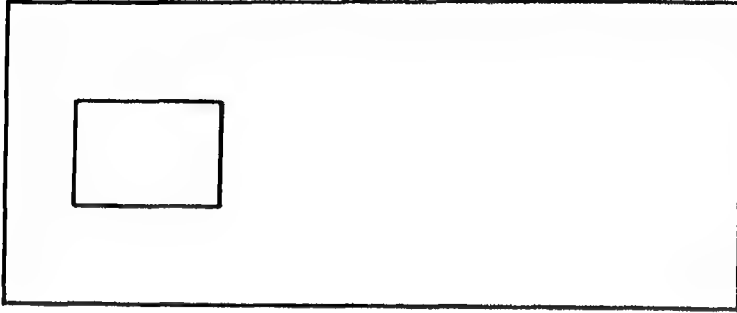
وباستخدام تعليمة خط LINE يمكننا رسم أشكال رباعية ، حيث إن كل ما هو مطلوب لرسم الأشكال الرباعية تحديد إحداثيات زاويتي (نقطتين) متعاكستين ، واستخدام حرف B اختصاراً لكلمة صندوق BOX مقروناً بنهاية تعليمة خط ، كما هو مبين بالشكل التالي :

الشكل العام لعبارة تعليمة خط LINE لرسم أشكال رباعية					
XXX	LINE	(N1, N2)	—	(N3, N4)	N, B
رقم	تعليمة	إحداثيات الزاوية	إحداثيات الزاوية	رمز لون	لتحديد أن
السطر	خط	الأولى	المعاكسة	خطوط الشكل	الشكل رباعي

فبالعبارة التالية

10 LINE (10,30) - (70,100), 1, B

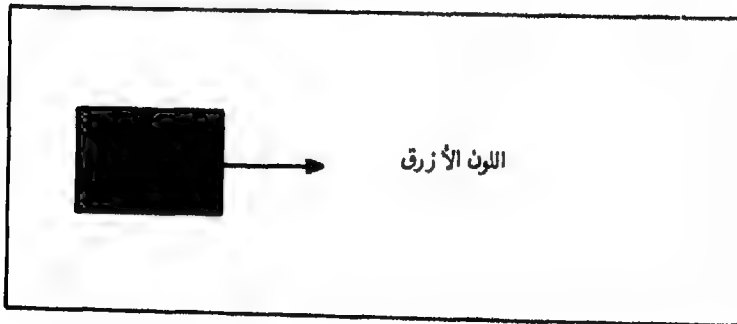
نعمل على رسم شكل رباعي وإحداثيات زاويته الأولى (10, 30) ، والزاوية المعاكسة (70, 100) ، ولون الخطوط سيكون أزرق كما هو مبين فيما يلي :



كما يمكننا تلوين هذا الشكل الرباعي بكامله من الداخل باستخدام حرف F اختصاراً لكلمة FILLED ملاصقة لحرف B في نهاية العبارة . فإذا عدلنا العبارة السابقة لتصبح :

10 LINE (10,30)-(70,100),1,BF

فإن الشكل الرباعي يصبح كالتالي :



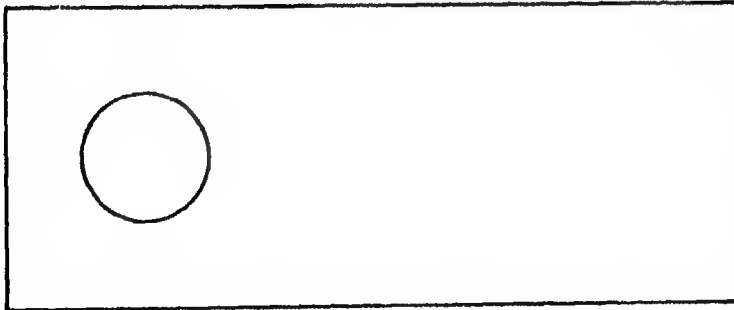
الدائرة : في الرياضيات (علم الهندسة) ، يمكننا رسم الدائرة إذا عرفنا (القطر أو نصف القطر) وإحداثيات مركز الدائرة، أما في لغة البيسك فيجب معرفة طول نصف القطر وإحداثيات المركز، ومن ثم يمكننا رسم الدائرة باستخدام تعليمة دائرة CIRCLE ، كما هو مبين بالشكل التالي :

الشكل العام لعبارة تعليمة دائرة				
CIRCLE				
XXX	CIRCLE	(N1, N2)	, N,	N
↓	↓	↓	↓	↓
رقم السطر	تعليمة دائرة	إحداثيات المركز	طول نصف القطر	لون خط الدائرة

فبالعبارة التالية :

10 CIRCLE (80, 80), 40, 1

نعمل على رسم دائرة إحداثيات مركزها (80,80) ونصف قطرها يساوى 40 ولون الخط الذى ستظهر به أزرق، كما هو مبين فيما يلى :



لاحظ وجود التعرجات في الخط الدائري، و يعود ذلك إلى الحالة التي توجد بها الشاشة وهي متوسطة الجودة SCREEN 1 وإذا عملنا على تحويل الحالة إلى عالية الجودة SCREEN 2 فإن هذه التعرجات ستقل و يظهر الخط الدائري متلاصقاً وأفضل بكثير مما كان عليه .

القوس : إن الزوايا تقاس بعدد درجاتها، فمثلاً نقول إن الزاوية أ ب ج تساوي ٩٠ درجة، أى زاوية قائمة، والزاوية المستقيمة تساوي ١٨٠ درجة، والزاوية الكاملة تساوي ٣٦٠ درجة، و يستخدم الباي PI (وهو الحرف السادس عشر من الأ بجدية اليونانية والذي يحمل القيمة العددية 3,1415 و يرمز له بالشكل π في تمثيل النسبة ما بين طول محيط الدائرة وقطرها .

ولرسم جزء من الدائرة نستخدم تعليمة دائرة، كما هو مبين بالشكل التالي :

الشكل العام لعبارة تعليمة دائرة لرسم جزء من الدائرة						
xxx	CIRCLE	(N1, N2),	N ₁	N ₂	N ₃	N
↓	↓	↓	↓	↓	↓	↓
رقم السطر	تعليمة دائرة	إحداثيات مركز الدائرة	نصف القطر	اللون	بداية الزاوية	نهاية الزاوية

فبالعبارة التالية :

10 CIRCLE (80,80), 40,1,,1,1.25

نعمل على رسم قوس أو جزء من الدائرة، كما هو مبين فيما يلي :



ولربط كل من بداية ونهاية القوس بمركز الدائرة نضع إشارة سالبة أمام كل من بداية ونهاية الزاوية .

فبالعبارة التالية :

10 CIRCLE (80,80), 40, 1,-,1,-1.25

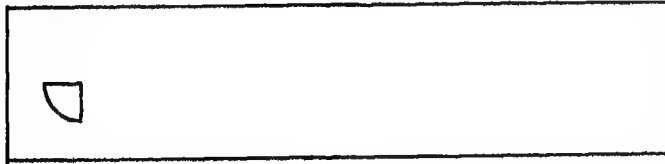
نعمل على رسم جزء من الدائرة تتصل بمركزها، كما هو مبين فيما يلي :



أما العبارة التالية :

10 CIRCLE (80,80), 40,1,-PI,-3 * (PI / 2)

فتعمل على رسم الربع الأيسر السفلي من الدائرة، كما هو مبين فيما يلي :



تطوير برنامج باستخدام تعليمة الدائرة CIRCLE

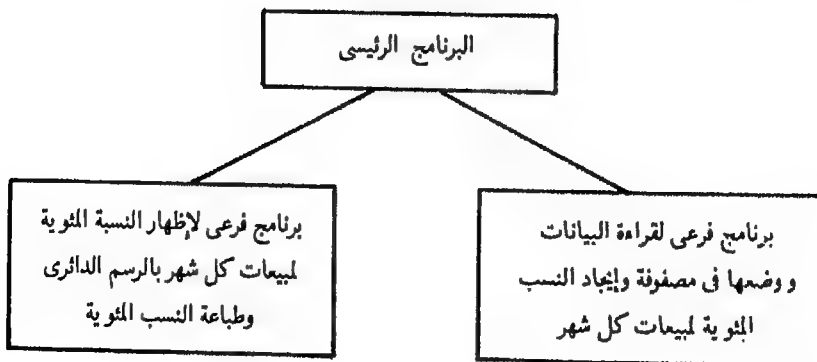
مثال (١٦ - ٣) : (الدائرة المقسمة)

الهدف : تمثيل مبيعات الشركة للاثني عشر شهراً (مثال ١٦ - ١) بالرسم الدائري المقسم (PIE CHART) .

أولاً - خطوات الحل :

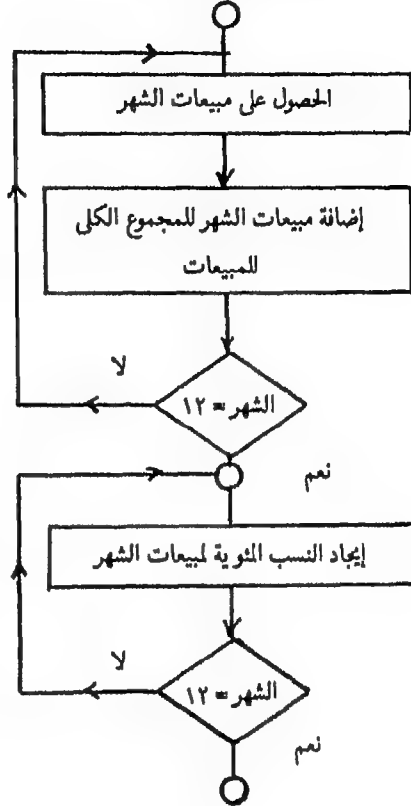
- ١ - الحصول على مبيعات الشهور الاثني عشر ووضعها في مصفوفة .
- ٢ - إيجاد المجموع الكلي لمبيعات الشركة خلال الاثني عشر شهراً .
- ٣ - إيجاد النسبة المئوية لمبيعات كل شهر .
- ٤ - تمثيل النسبة المئوية لكل شهر حسب قيمتها في الدائرة .
- ٥ - تكرار الخطوات ٣ و ٤ لجميع الشهور الاثني عشر .
- ٦ - طباعة النسب المئوية لجميع الشهور .

ثانياً - الهيكل الهرمي :

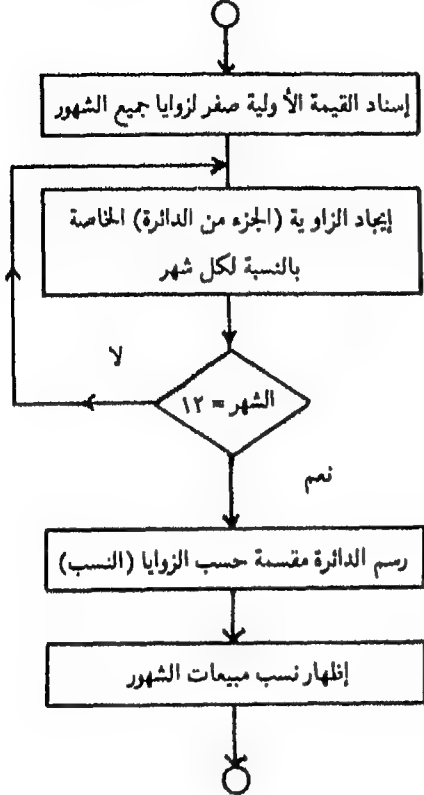


ثالثاً - فط البرمجة التركيبية :

قراءة البيانات وإيجاد النسب



تمثيل النسب المئوية بالرسم الدائري



رابعاً - البرنامج في الشكل التالي :

شكل (١٦-٣)

برنامج لإظهار مبيعات الشركة لاثني عشر شهراً برسم الدائرة المثلثة

```

10 C!S: REM      الشاشة
20 SCREEN 2,1,0
30 DIM T(20),S(12),P(12),A(12)
40 GOSUB 70 :REM      لقراءة البيانات
50 GOSUB 210 :REM      لمخرجات
60 END
70 REM      برنامج فرعي
75 REM      الشهر
80 LET T=0
90 FOR I = 1 TO 12
100 READ S(I)
110 LET T = T + S(I) :REM      المبيعات
120 NEXT I
130 REM      إضافة مبيعات الشهر لمجموع المبيعات
140 FOR I = 1 TO 12
150 LET P(I) = (S(I) / T)
160 NEXT I
170 REM      البيانات
180 DATA 6000,10000,13000,7500,10500,14000
190 DATA 9300,3500,10000,5000,11000,14000

```

```

200 RETURN
210 REM          البرنامج الفرعي للظهار مبيعات كل شهر بالرسم الدائري
220 FOR I=0 TO 12 : REM          لاسناد القيمة الاولي لزاوية كل شهر
230 LET A(I) = 0
240 NEXT I
250 REM          لليجاد الزاوية لمبيعات كل شهر
260 FOR I = 1 TO 12
270 LET D = P(I) * (2 * 3.141)
280 LET A(I) = A(I - 1) + D
290 CIRCLE (400,100),80,1,-A(I),-A(I-1)
300 NEXT I
310 REM          لمطابقة النسبة المئوية لمبيعات كل شهر
320 LOCATE 1,1: PRINT "النسبة المئوية لمبيعاته"
330 LOCATE 2,1: PRINT "=====الشهر====="
340 FOR I=1 TO 12
350 PRINT USING "      #.###"      ## " :P(I), I
360 NEXT I
370 RETURN

```


وفي حالة تنفيذ البرنامج سنحصل على النتائج التالية المطبوعة :

النسبة المئوية لمبيعات	الشهر
=====	=====
0.053	1
0.088	2
0.114	3
0.066	4
0.092	5
0.123	6
0.082	7
0.031	8
0.088	9
0.044	10
0.097	11
0.123	12

أما الدائرة، فيمكن مشاهدتها على الشاشة .

تمارين

١ — اعمل على طباعة وتنفيذ البرامج الموجودة في أشكال الفصل ، وإجراء التعديلات عليها إن وجدت ، وتنفيذها حسب تسلسل هذه التعديلات .

٢ — بين الأخطاء إن وجدت في العبارات التالية :

- | | | |
|----|---------------------------|----|
| 10 | CIRCLE (20, 30) | أ |
| 10 | LENE (2,3) - (40,70),1,BN | ب |
| 10 | LINE-STEP (70,-30) | ج |
| 10 | SCREEN 7,40,20 | د |
| 10 | PI = 4.314159 | هـ |

٣ — اعمل على تعديل العبارات وإجراء البرامج التالية لكي يتم تنفيذها بدون أخطاء :

- | | | |
|----|----------------------------|--|
| 10 | LINE (20,80) (70,120),1,B | أ — لرسم مستطيل |
| 10 | CIRCLE (50,50) 20,1,.1,1.5 | ب — لرسم قوس |
| | | ج — لرسم خطوط توصل زوايا الشاشة الأربع |

```
10 FOR I = 1 TO 80
20 LOCATE 5,I: PRINT CHR$(196)
30 NEXT I
40 FOR I = 1 TO 25
50 LOCATE 1,I: PRINT CHR$(196);
60 NEXT I
```

٤ — اكتب العبارات اللازمة لرسم كل مما يلي :

أ — دائرة نصف قطرها ٣٠ وإحداثيات المركز (٦٠ ، ٥٠) .

ب — مثلث (أ ب ج) مع العلم بأن إحداثيات أ = (٢٠ ، ٣٠) ، ب = (٥٠ ، ٦٠) ، ج = (٩٠ ، ١٠٠) .

ج — اعمل على تعديل كل من أ ، ب بحيث تتم تغطية الأشكال من الداخل (تظليلها) .

٥ — في التمرين رقم ٤ في الفصل الثالث عشر، أظهر النتائج باستخدام النقاط المبعثرة

SCATTERDIAGRAM

أ — رسم بالقيم الفعلية للأسهم .

ب — رسم لنسب التغيير .

٦ — يتسلم المركز الرئيسى لإدارة محلات بنده فى الرياض تقريراً فى نهاية كل يوم ، فيه

إجمالى المبيعات لكل فرع ، يقوم محمد — السكرتير — كل أسبوع بتزويد المدير

العام بتقرير فى صباح يوم السبت يحتوى على مجموعات المبيعات الأسبوعية لكل

فرع من الفروع ونسبة مبيعات كل فرع إلى المجموع الكلى .

١ — طور برنامجاً للقيام بهذه المهمة علماً بأن هناك خمسة فروع لبندة فى الرياض .

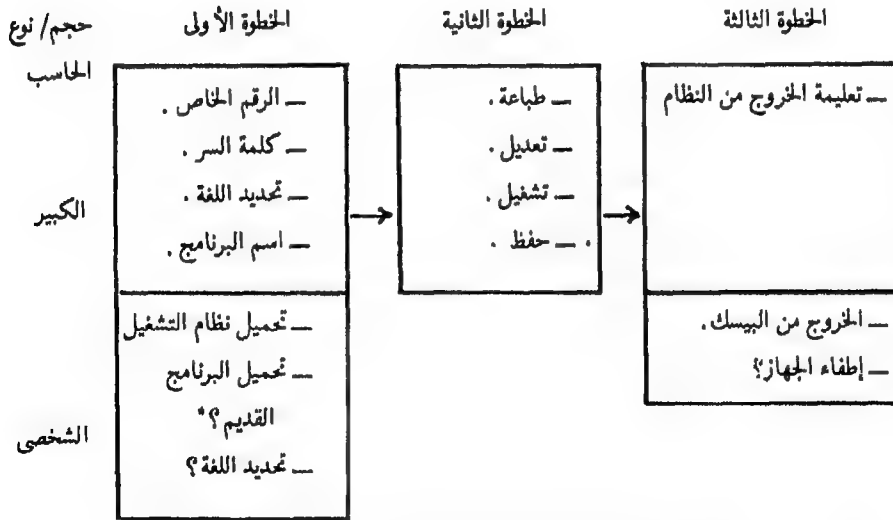
٢ — أظهر النتائج بواسطة الدائرة المقسمة .

ملحق عمليات التشغيل

مقدمة عن عمليات التشغيل

يمكن تقسيم الخطوات المتبعة في التخاطب مع الحاسبات بأحجامها المختلفة إلى ثلاث خطوات رئيسية :

- (١) عملية الدخول إلى الحاسب وتعريف المستخدم واللغة .
 - (٢) عملية التخاطب مع الحاسب مثل طباعة وتغيير البرامج وتشغيلها .
 - (٣) عملية الانتهاء والخروج من الحاسب .
- ويبين الرسم التالى تسلسل هذه الخطوات وأنواع الأنشطة والمعلومات المطلوبة لكل خطوة وذلك حسب حجم الحاسب .



• ؟ : إذا اقتضى الأمر ذلك .

لاحظ عدم وجود فروقات في أنشطة الخطوة الثانية بين الحاسبات الآلية الكبيرة والصغيرة، والفروقات في الخطوتين الآخرين ناتجة عن أن الحاسبات الكبيرة هي متعددة المستخدمين، وأن الحاسبات الشخصية — في الغالب — هي ذات مستخدم واحد. وتعدد المستخدمين يؤدي إلى تعدد مقاصد الاستخدام ولغات التخاطب مع الحاسب.

ويعنى هذا أمرين :

١ — أن يوضع الحاسب في حالة الاستعداد الدائم لقبول طلبات المستخدمين .

٢ — تعريف المستخدمين ونوع لغة التخاطب .

وسنبحث في بقية الفصل الخطوات المحددة للتخاطب مع نوعين من الحاسبات الكبيرة IBM, HP-3000 وثلاثة أنواع من الحاسبات الشخصية :

آى. بى. أم IBM وراديو شاك RADIO-SHACK وأبل APPLE

خطوات التعامل مع الحاسبات الكبيرة

إن أسس خطوات التعامل مع الحاسبات الآلية الكبيرة MAIN FRAME COMPUTERS واحدة حتى مع اختلاف نوعية هذه الأجهزة، ففي جميع المؤسسات والشركات يوجد لكل مستخدم رقم خاص USER-ID وكلمة سر PASSWORD خاصة به، حيث يقوم كل مستخدم بإدخالهما ومن ثم يعمل نظام التشغيل باستدعاء برنامج خاص (مخزن به جميع الأرقام الخاصة بالمستخدمين وكلمة السر لكل رقم)؛ ليعمل على التأكد من صلاحية الرقم وكلمة السر للمحافظة على بيانات وأجهزة المركز، ففي حالة عدم تطابقهما يعمل الجهاز على إظهار رسالة بذلك، وفي معظم الحالات لايسمح لهذا المستخدم بالتعامل مع الجهاز.

وستعرض الآن لخطوات الدخول والتعامل مع بعض هذه الأجهزة :

أ) جهاز آى بى إم I. B. M

١ - يتم التعامل مع معظم لغات البرمجة باستخدام نظام TSO وذلك بكتابة كلمة TSO ومن ثم الضغط على مفتاح العودة/ الإدخال ENTER .

٢ - سيسأل المستخدم عن إدخال رقمه بالرسالة التالية ENTER USERID .

٣ - بعد كتابة الرقم والضغط على مفتاح العودة/ الإدخال ENTER سيسأل المستخدم عن إدخال كلمة السر بالرسالة التالية

. ENTER CURRENT PASSWORD FOR USERID

٤ - بعد كتابة كلمة السر والضغط على مفتاح العودة/ الإدخال يتم التدقيق فى صلاحيتهما، وفى حالة السماح للمستخدم باستعمال الجهاز ستظهر كلمة READY لإشعار المستخدم بأن الجهاز فى حالة الاستعداد لاستقبال التعليمات .

٥ - لكتابة برنامج بلغة بيسك تكتب الجملة التالية :

EDIT TEST (XXXXXX) VSBASIC

↑
اسم البرنامج

ومن ثم الضغط على مفتاح العودة/ الإدخال .

ففى حالة كون البرنامج جديداً ستظهر كلمة INPUT وفى سطر آخر يظهر رقم 0010 للبداية فى كتابة جمل البرنامج، وفى حالة الانتهاء من جميع جمل البرنامج والضغط على مفتاح العودة/ الإدخال ستظهر كلمة EDIT أى أن بإمكان المستخدم إجراء أى من العمليات المبينة فى جدول (ب)، أما إذا كان البرنامج قد خزن سابقاً فسيتم تحميله تلقائياً .

٦ - وبعد الانتهاء من طباعة البرنامج أو تعديله يمكن حفظه بإحدى وسائل التخزين باستخدام العبارة التالية END SAVE ومن ثم الضغط على مفتاح العودة/ الإدخال، بعد ذلك ستظهر كلمة READY .

٧ - للخروج من النظام نكتب تعليمة LOGOFF ومن ثم الضغط على مفتاح العودة/ الإدخال .

ب) جهاز هيوليت باكرد ٣٠٠٠ HEWLETT PACKARD 3000 SERIES III

١ - بالضغط على مفتاح العودة/ الإدخال يتم وضع الحاسب في حالة الاستعداد لاستقبال التعليمات من المستخدم، وذلك بعد إظهار نقطتين «فوق بعض» : COLON .

٢ - بعد إظهار النقطتين تتم طباعة السطر التالى :

: HELLO XXXXX ; TERM = 6

رقم

ومن ثم الضغط على مفتاح العودة/ الإدخال .

٣ - بعد إدخال رقم المستخدم والتأكد من صحته سيطلب من المستخدم إدخال كلمة السر الخاصة به، وبعد إدخالها والضغط على مفتاح العودة/ الإدخال سيتم التأكد من صحتها أيضاً، وفي حالة صحتها سيتم إظهار عدة رسائل على الشاشة، بعدها يتم إظهار نقطتين «فوق بعض» (: COLON) لإشعار المستخدم بأن الجهاز في حالة الاستعداد لاستقبال التعليمات .

٤ - للتعامل مع لغة بيسك تكتب كلمة BASIC ومن ثم الضغط على مفتاح العودة/ الإدخال . سيظهر بعد ذلك إشارة أكبر من > لإشعار المستخدم بأن الجهاز في حالة الاستعداد لتلقى التعليمات الخاصة بلغة بيسك .

٥ - لإدخال برنامج جديد نبدأ بكتابة العبارات حسب قواعد وتعليمات لغة بيسك والضغط على مفتاح العودة/ الإدخال بعد الانتهاء من كل عبارة . أما لاستدعاء برنامج قديم فنستخدم العبارة اسم البرنامج GET XXX .

٦ - في حالة الانتهاء من كتابة/ تعديل/ تنفيذ البرنامج يمكن حفظه بإحدى وسائل التخزين بكتابة تعليمة احفظ واسم البرنامج كالتالى: اسم البرنامج XXX

SAVE ومن ثم الضغط على مفتاح العودة/ الإدخال . وبعد ذلك ستظهر إشارة أكبر من > أى أن بإمكان المستخدم التعامل مع أى برنامج من برامج لغة بيسك بإجراء أى من العمليات المبينة فى جدول (أ) .

٧ - للخروج من نظام بيسك إلى نظام التشغيل نكتب تعليمة نظام SYSTEM ومن ثم الضغط على مفتاح العودة/ الإدخال ، ستظهر بعد ذلك نقطتان «فوق بعض» (:). ولإنهاء التعامل مع الجهاز نكتب تعليمة BYE ومن ثم الضغط على مفتاح العودة/ الإدخال .

خطوات التعامل مع الحاسبات الآلية الشخصية

إن خطوات التعامل مع الحاسبات الآلية الشخصية PERSONAL COMPUTERS تختلف إلى حد ما عن التعامل مع الحاسبات الآلية الكبيرة؛ ففى الحاسبات الكبيرة يعمل المشغل THE OPERATOR على تجهيز الحاسب ووضعه فى حالة الاستعداد للمستخدمين وذلك بتحميل نظام التشغيل للذاكرة الرئيسية، أما فى حالة الحاسبات الشخصية، والتى يطلق عليها فى بعض الأحيان اسم الحاسبات الصغيرة SMALL COMPUTERS، فيقوم المستخدم بتجهيز الجهاز وتحميل نظام التشغيل للذاكرة الرئيسية باتباع الخطوات التالية :

- (أ) وضع الأسطوانة التى تحتوى على نظام التشغيل و يطلق عليها اسم DISK OPERATING SYSTEM (DOS) فى وحدة الأسطوانة الرئيسية DISK DRIVE والتى عادة ما تحمل الرقم 1 أو الحرف A .
- (ب) توصيل التيار الكهربائى وذلك بوضع مفتاح الكهرباء فى حالة ON .

وسنتعرض الآن لكيفية التعامل مع لغة بيسك باستخدام ثلاث حاسبات آلية
شخصية مختلفة :

أولاً — جهاز آى بى إم I.B.M

١ — بعد تنفيذ الخطوتين (أ، ب) سيبدأ البحث عن نظام التشغيل DOS ومن ثم
تحميله للذاكرة الرئيسية . بعد ذلك سيطلب من المستخدم إدخال التاريخ
والوقت .

ففى حالة إدخال التاريخ حسب الصيغة المظهرة سيحل فى الذاكرة الرئيسية
مكان التاريخ الأصلى ، ويمكننا الضغط على مفتاح العودة وإبقاء التاريخ القديم
دون أن يؤثر على سير العمل .

٢ — بعد الانتهاء من إدخال التاريخ والوقت سيظهر حرف A متبوعاً بإشارة أكبر من (> A) وهذا يعنى أن الجهاز فى حالة الاستعداد لاستقبال أى تعليمات والتعامل مع
الأسطوانة الموجودة فى الوحدة A .

٣ — للتعامل مع نظام البرمجة بلغة بيسك يمكننا استدعاء نظام BASIC أو نظام *
BASICA (وهو عبارة عن نظام BASIC المعدل بإضافة بعض التعليمات
المتقدمة) . ويتم ذلك بكتابة مترجم اللغة المتوفر ومن ثم الضغط على مفتاح
العودة/ الإدخال (↵) .

٤ — بعد الاستدعاء سيتم البحث عن النظام وتحميله للذاكرة الرئيسية وبعد الانتهاء
من ذلك ستظهر الرسائل التالية :

THE IBM PERSONAL COMPUTER BASIC

VERSION 2.1 COPYRIGHT IBM., CORP., 1981, 1982, 1983

61327 BYTES FREE

• سيتم شرح خطوات التعامل مع BASICA فى الجزء الأخير من هذا الفصل .

- ٥ — هنا يكون الجهاز في حالة الاستعداد للتعامل مع لغة بيسك . فإذا أردنا إدخال برنامج نبدأ بكتابة العبارات والضغط على مفتاح العودة/ الإدخال بعد الانتهاء من كل عبارة إلى أن ننتهي من كتابة البرنامج .
- ٦ — يمكننا التعامل مع هذا البرنامج باستخدام أى من العمليات المبينة في جدول (أ) .

ثانياً — جهاز أبل APPLE

- ١ — بعد تنفيذ كل من الخطوتين (أ، ب) سيبدأ البحث عن نظام التشغيل DOS ومن ثم تحميله للذاكرة الرئيسية، وبعد الانتهاء ستظهر الرسائل التالية على الشاشة :
- ```
DOS VERSION ## * ## ## / ## ## / ## ## **
APPLE II PLUS OR ROMCARD SYSTEM MASTER
(LOADING INTEGER INTO LANGUAGE CARD)
```
- ٢ — بعد ظهور المؤشر CURSOR لإشعار المستخدم بأن الجهاز في حالة الاستعداد لاستقبال التعليمات، يجب كتابة كلمة جديد NEW لحذف أى برنامج يتعلق بلغة بيسك ومحمل بالذاكرة الرئيسية .
- ٣ — بعد الانتهاء من كتابة NEW والضغط على مفتاح العودة سيصبح الجهاز في حالة الاستعداد لاستقبال أى تعليمة تتعلق بلغة بيسك .
- ٤ — إذا أردنا إدخال برنامج جديد نبدأ بكتابة العبارات، ومن ثم الضغط على مفتاح العودة بعد الانتهاء من كل عبارة إلى أن ننتهي من كتابة البرنامج .
- ٥ — بعد ذلك يمكننا التعامل مع هذا البرنامج باستخدام أى من التعليمات المبينة في جدول (أ) .

٥ رقم نسخة نظام التشغيل .  
 ٥٥ التاريخ .

١ — بعد تنفيذ كل من الخطوتين (أ، ب) سيبدأ البحث عن نظام التشغيل DOS ومن ثم تحميله للذاكرة الرئيسية .

٢ — بعد الانتهاء من مرحلة التحميل سيظهر المؤشر على الزاوية اليسرى العلوية من الشاشة، وعندها يكون الجهاز في حالة الاستعداد لتلقى التعليمات من المستخدم .

٣ — للتعامل مع لغة بيسك ندخل التعليمات BASIC، ومن ثم الضغط على مفتاح العودة/ الإدخال، بعد ذلك ستظهر الرسائل التالية :

RADIO SHACK MODEL III BASIC

(C)' 80 Tandy

READY

>

٤ — هنا يكون الجهاز في حالة الاستعداد للتعامل مع لغة بيسك، فإذا أردنا إدخال برنامج نبدأ بكتابة العبارات والضغط على مفتاح العودة/ الإدخال بعد الانتهاء من كل عبارة إلى أن ننتهي من كتابة البرنامج .

٥ — يمكننا التعامل مع هذا البرنامج باستخدام أى من التعليمات المبينة في جدول (أ) .

جدول (١٧-أ)

بعض العمليات والتعليمات المتعلقة بكل جهاز في الحاسبات الآلية الشخصية

| RADIO SHACK  | جهاز راديو شاك | APPLE         | جهاز أبل      | IBM           | جهاز آي بي إم | نوع العملية                                                                                                                                                                                                                                                                |
|--------------|----------------|---------------|---------------|---------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LIST (ENTER) | LIST (ENTER)   | LIST (RETURN) | LIST (RETURN) | LIST (RETURN) | LIST (RETURN) | ١- لإظهار البرنامج على الشاشة<br>٢- لتنفيذ البرنامج.<br>٣- حفظ البرنامج.<br>٤- تحميل البرنامج.<br>٥- للحصول على نسخة مطبوعة من البرنامج.<br>٦- لطباعة النتائج.                                                                                                             |
| RCN          | RCN            | RCN           | RCN           | RCN           | RCN           | ٧- لإظهار جزمين "برنامج على الشاشة.<br>٨- لطباعة جزء من "برنامج.<br>٩- مسح الشاشة وإعادة المؤشر الزاوية اليسرى العلوية.<br>١٠- حذف عبارة من البرنامج.<br>١١- حذف أكثر من عبارة من البرنامج.<br>١٢- حذف البرنامج من وحدة التخزين.<br>١٣- العودة من نظام يسلك لنظام التشغيل. |
| SAVE         | SAVE           | SAVE          | SAVE          | SAVE          | SAVE          |                                                                                                                                                                                                                                                                            |
| LOAD         | LOAD           | LOAD          | LOAD          | LOAD          | LOAD          |                                                                                                                                                                                                                                                                            |
| LLIST        | PR#0<br>LIST   | PR#0<br>LIST  | PR#0<br>LIST  | PR#0<br>LIST  | PR#0<br>LIST  |                                                                                                                                                                                                                                                                            |
| LPRINT       | PR#1<br>PRINT  | PR#1<br>PRINT | PR#1<br>PRINT | PR#1<br>PRINT | PR#1<br>PRINT |                                                                                                                                                                                                                                                                            |
| LIST         | LIST           | LIST          | LIST          | LIST          | LIST          |                                                                                                                                                                                                                                                                            |
| LLIST        | PR#0<br>LIST   | PR#0<br>LIST  | PR#0<br>LIST  | PR#0<br>LIST  | PR#0<br>LIST  |                                                                                                                                                                                                                                                                            |
| GLS          | HOME           | HOME          | HOME          | HOME          | HOME          |                                                                                                                                                                                                                                                                            |
| DELETE       | DELETE         | DELETE        | DELETE        | DELETE        | DELETE        |                                                                                                                                                                                                                                                                            |
| DELETE       | DELETE         | DELETE        | DELETE        | DELETE        | DELETE        |                                                                                                                                                                                                                                                                            |
| TRSDOS       | ESC            | ESC           | ESC           | ESC           | ESC           |                                                                                                                                                                                                                                                                            |

جدول (١٧ - ب)

| بعض العمليات والتعليمات المتعلقة بكل منها في الحاسبات الآلية الكبيرة                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HP - 3000 SERIES III                                                                                                                                                                                                                                                                                                                                                                 | L. B. M. جهاز آتومي إم                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | نوع العملية                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <p>LIST<br/>RUN<br/>SAVE<br/>GET اسم البرنامج<br/>طابعة رقم العبارة<br/>إلى رقم عبارة - من رقم عبارة<br/>PURGE اسم البرنامج<br/>اسم البرنامج الجديد واسم البرنامج القديم<br/>RENAME<br/>: FILE PRINT; DEV = LP<br/>: BASIC<br/>&gt; GET البرنامج اسم<br/>&gt; LIST, OUT = PRINTER<br/>: FILE PRINTER; DEV = LP<br/>: BASIC<br/>&gt; GET البرنامج اسم<br/>&gt; RUN, OUT = PRINTER</p> | <p>LIST<br/>RUN<br/>END SAVE<br/>EDIT TEST (اسم البرنامج) VS BASIC<br/>طابعة رقم العبارة<br/>إلى رقم عبارة - من رقم عبارة<br/>DELETE X «لرقم المستخدم»<br/>TEST. VSBASIC (اسم البرنامج)<br/>رقم المستخدم RENAME<br/>TEST. VSBASIC (اسم البرنامج)<br/>TEST. VSBASIC (اسم البرنامج)<br/>رقم المستخدم TEST. VSBASIC (اسم البرنامج)<br/>اسم الطابعة (اسم البرنامج) TEST. VSBASIC<br/>// STEROL EXEC PGM = ICDO5 BSC<br/>// SYSPRINT DD SYSOUT = *<br/>// CONTROL DD *<br/>RUN ABC SOURCE<br/>// *<br/>// ABC DD *<br/>// [البرنامج]<br/>//<br/>//</p> | <p>١- إظهار البرنامج على الشاشة.<br/>٢- تنفيذ البرنامج.<br/>٣- حفظ البرنامج.<br/>٤- تحميل البرنامج.<br/>٥- حذف عبارة من البرنامج.<br/>٦- حذف أكثر من عبارة من البرنامج.<br/>٧- حذف البرنامج من وحدة التخزين.<br/>٨- تغيير اسم البرنامج.<br/>٩- للحصول على نسخة مطبوعة من البرنامج.<br/>١٠- للحصول على النتائج مطبوعة في الحاسبات الآلية الكبيرة وخاصة آتومي إم، نحتاج إلى لغة التحكم في التشغيل JCL وفي الحصول ل Job Control Card</p> |

## الإجراءات الخاصة بطباعة البرامج المطورة في الكتاب

تم تطوير جميع البرامج باستخدام جهاز آي بي إم IBM الشخصي العادي ، سعة الذاكرة ٢٥٦ ألف حرف (256 K) واستخدام لغة BASICA في كتابة البرامج والتي تسمح بالتعامل مع البيانات في اللغتين العربية والإنجليزية .

وللتعامل مع BASICA لابد من توفر النظم المتضمنة في ASD والتي تكون مخزنة على قرص نظام التشغيل وتتبع الخطوات التالية في عملية التشغيل :

١ — تحميل قرص نظام التشغيل والذي يحتوى على البرامج التالية :

أ) WTDATIM ويتم تحميله تلقائياً بعد تشغيل الجهاز .

ب) ARABIC ويتم إدخالها من قبل المستخدم ، ومن ثم الضغط على مفتاح الإدخال/العودة لتحميل الحروف والأرقام والرموز العربية في الذاكرة الرئيسية ، وبعد إتمام ذلك ستظهر الرسالة التالية :

ASD OPTION SPECIFIED AND LOADED

ج) AMODE A ويتم إدخالها من قبل المستخدم ومن ثم الضغط على مفتاح الإدخال/العودة لتعريف الحروف والأرقام والرموز العربية المظهرة على لوحة المفاتيح .

٢ — طباعة BASICA لتحميل مترجم لغة بيسك .

٣ — يتم طباعة تعليمات بيسك باللغة الانجليزية .

٤ — للتحويل من حالة اللغة الإنجليزية إلى حالة اللغة العربية ، من أجل طباعة العناوين أو الرسائل أو البيانات أو الشرح والتعليق ، تتبع الخطوات التالية :

أ) الضغط المستمر على مفتاح (+ALT) ومن ثم الضغط على المفاتيح التاليين

بالتوالى (↑) الموجودين في الجهة اليمنى من لوحة المفاتيح ومفتاح (↵) الموجود في  
الجهة العلوية اليسرى من لوحة المفاتيح .

وفي حالة إتمام هذه العمليات بنجاح يصدر الحاسب صفيراً قصيراً . وفي حالة عدم  
سماع الصفير تتم إعادة هذه الخطوة .

ب ( للعودة إلى حالة اللغة الإنجليزية يتم الضغط المستمر على مفتاح (ALT) ومن ثم  
الضغط على مفتاح (↑) الموجود في الجهة اليسرى من لوحة المفاتيح .

## مقارنة لغة بيسك بلغتى البرمجة شائعتى الاستخدام (فورتران ، كوبول)

ستكون أسس المقارنة وفق العوامل الأساسية التالية :

### أولاً - الطبيعة العامة للغة :

تحديد الهدف الأساسى من تطوير اللغة ، طبيعتها العامة وذلك كالتالى :  
أ - لغة بيسك : تستخدم لغة بيسك لتسهيل عملية التخاطب المباشر مع الحاسبات الشخصية والتي استخدمت بشكل خاص فى المؤسسات التعليمية ، وكلمة بيسك BASIC باللغة الإنجليزية هى الأحرف الأولى من الكلمات التالية :

Beginners All-purpose Symbolic Instruction Code.

ب - لغة فورتران : تلائم لغة فورتران توصيف العمليات الحسابية المعقدة كتلك التى فى التطبيقات العلمية المجردة ، وكلمة فورتران FORTRAN باللغة الإنجليزية مقتبسة من الكلمتين

FORMAL TRANSLATOR

ج - لغة كوبول : تناسب لغة كوبول التطبيقات التجارية والتى تتطلب إنتاج تقارير دون الاعتماد على عمليات حسابية معقدة ، وقد اشتقت كلمة كوبول COBOL من الكلمات التالية :

Common Business Oriented Language

### ثانياً - تركيبة البرنامج :

بناء على الطبيعة العامة للغة نجد أن درجة تركيبة البرنامج تتفاوت ما بين اللغات الثلاث ، ف لغة بيسك تعتبر من أكثر اللغات سهولة حيث لا تتبع تركيباً معيناً ، أما لغة



فورتران فلها بعض القوانين المحددة وبشكل خاص تلك التي توضح كيفية إدخال البيانات وإظهارها ، أما لغة كوبول فهي أكثر اللغات تقيداً بقوانين محددة لأجزاء البرنامج ، حيث إن كل برنامج يحتوى على أجزاء معينة بغض النظر عن الهدف منه .

### ثالثاً — أسس البرمجة الهيكلية والتركيبية :

ذكرنا في المقدمة عاملين أساسيين يسهل توفرهما اتباع أساليب التجزئة الهرمية والبرمجة التركيبية :

أ ) تحديد بدايات ونهايات الأجزاء في البرنامج الواحد .

ب ) التحكم في تكرار تنفيذ أجزاء البرنامج .

لايوجد هناك اختلاف كبير بين اللغات الثلاث في مدى قابليتها لتوفير العاملين المذكورين آنفاً ، ولكن قد نجد أن اتباع الأساليب الحديثة في البرمجة ميسر في لغة كوبول أكثر منها في اللغتين الأخريين .

### رابعاً — سهولة فهم البرامج المكتوبة باللغات :

نجد أن لغة كوبول هي أقرب لغات البرمجة إلى اللغة الإنجليزية العادية ، لذلك نرى ضرورة تناثر العبارات التوضيحية في لغتي بيسك وفورتران .

وفيما يلي جدول تفصيلي يبين المقارنة بين اللغات الثلاث وفق بعض العمليات :

مقارنة بين التعليمات والقواعد الخاصة بها في لغات البرجة (بيك، فورتران وكوبول)

جدول (١٧- ج)

| نوع العملية                                                                                                              | لغة بيك                                         | لغة فورتران                                                                                                                                            | لغة كوبول                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| أسماء المتغيرات<br>VARIABLE NAMES                                                                                        | حرف أو حرف متبوع برقم<br>A٩ ... A٩ / A٩, A٩, A٩ | من حرف إلى ستة أحرف<br>C, COUNT, COLUNT 4                                                                                                              | من حرف إلى ٣٠ حرفاً<br>READ, PRINT - ROUTINE                                                                                             |
| تعريف البيانات واختيار<br>أسماء المتغيرات لها<br>وتوصيفها<br>DATA TYPES                                                  | قيم عديدة<br>N = قيم غير عديدة<br>NS            | — قيم عديدة صحيحة = يجب<br>أن يبدأ اسم المتغير بأي من<br>الحروف التالية I, J, K, L, M, N<br>— قيم عديدة غير صحيحة وغير<br>عديدة أي من الحروف A-H و O-Z | — قيم عديدة صحيحة PIC 999<br>— قيم عديدة غير صحيحة PIC 9V99<br>— قيم غير عديدة PIC XXX                                                   |
| عناوين البيانات وأجزاء<br>البرامج<br>LABELS<br>أعمدة ذات استخدام خاص<br>SPECIAL COLUMNS                                  | رقم سطر: 10....<br>20....<br>لا يوجد            | رقم 10<br>من ١ - ٥ لتعويض العبارات<br>٦ للكملة من السطر السابق                                                                                         | اسم يطلق على الجزء<br>READ - ROUTINE.<br>من A - ٣٧ أسماء الأجزاء<br>الرئيسية وأسماء العبارات.<br>١٢ - ٧٢ العبارات.<br>٧ للكملة أو الشرح. |
| الشرح والتعليق على العمليات<br>REMARKS / COMMENTS<br>آخر تعليمة في البرنامج<br>— لإنهاء العمليات في<br>البرنامج والتوقف. | نستخدم تعليمة REM                               | يستخدم الحرف C في السطور 1                                                                                                                             | يستخدم الرمز ه في السطور ٧<br>لم يحدد                                                                                                    |
| LAST STATEMENT                                                                                                           | END                                             | STOP                                                                                                                                                   | STOP - RUN                                                                                                                               |

|                                                                                                                     |                                                 |                                    |                       |                                                |
|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|------------------------------------|-----------------------|------------------------------------------------|
| MOVE 8 TO A.                                                                                                        | A = 8                                           | LET A = 8                          | MOVE/ASSIGN           | إستاد قيمة للمتغير                             |
| READ DATA-REC-IN ATEND<br>MOVE « NO » TO FLAG-EOD.<br>IF FLAG-EOD = «NO»PERFORM 500-END<br>CLOSE FILE-IN, FILE-OUT. | READ (5,10) A,N<br>10 FORMAT (F3,1,I2)          | READ A,N<br>DATA 5,14              | READING DATA          | قراءة البيانات                                 |
| WRITE DATA-REC-OUT AFTER<br>ADVANCING 2 LINES.                                                                      | WRITE (6,20) A,N<br>20 FORMAT (10X,F4,1,I4)     | PRINT A,N                          | PRINTING RESULTS      | طباعة نتائج العملية                            |
| ADD B TO C GIVING A.                                                                                                | A = B + C                                       | LET A = B + C                      | ARITHMETIC OPERATIONS | إجراء عمليات حسابية                            |
| A OCCURS 10 TIMES PIC 99.                                                                                           | DIMENSION A(10), B (5,20)                       | DIM A (10), B(5,20)                |                       | حجز أماكن للقيم في الذاكرة<br>(مصفوفات) ARRAYS |
| PERFORM 100-READ-AND-PROCESS<br>6 TIMES.                                                                            | DO 20 I = 1,6<br>:<br>:<br>20 CONTINUE          | FOR I = 1 TO 6<br>:<br>:<br>NEXT I | LOOPS                 | الدورة والتحكم في<br>تنفيذ الأجزاء             |
| IF A = 6 PERFORM 300-PRINT.                                                                                         | IF (A.EQ.6) GOTO 20                             | IF A = 6 THEN 300                  | COMPARISONS           | إجراء عملية مقارنة /<br>منطقية والتعريب        |
| DISPLAY « Enter the Value: »                                                                                        | WRITE (6,20)<br>20 FORMAT(10X, Enter the Value) | PRINT « Enter the Value : »        | MESSAGES INTERACTIVE  | طباعة الرسائل<br>(التخاطب مع المستخدم)         |

## المراجع العربية

### أ — معالجة البيانات :

- عوض منصور، محمد أبو النور، محمد العمري: «مقدمة في علم الحاسب الإلكتروني»، الأردن، ١٩٨٧.
- محمد الفيومي: «مقدمة في علم الحاسبات الإلكترونية ومعالجة البيانات»، دار الفرقان، الأردن، ١٩٨٤.
- محمد شوقي بشارة: «الحاسبات الإلكترونية ونظم المعلومات»، بيروت، ١٩٨٣.

### ب — البرمجة بلغة البيسك :

- عوض منصور: «برمجة بيسك للمبتدئين» الأردن، ١٩٨٧م.
- مظهر طایل: «الكمبيوتر لغة وأداء (بيسك)»، بيروت، ١٩٨٤.
- محمد الفيومي: «برمجة الحاسبات الإلكترونية بلغة بيسك»، دار الأمل، الأردن، ١٩٨٤.
- محمد السميد خشبة: «أساليب تخطيط البرامج بلغة البيسك»، جمهورية مصر العربية، ١٩٨٤.

## المراجع الأجنبية

### أ — معالجة البيانات :

- Basili, Victor R. and Baker, F. Terry, "Tutorial on Structured Programming and Integrated Practices," Los Alamos CA, IEEE Computer Science Press, 1981.
- Bohl, Marilyn, "Flowcharting Techniques," Chicago, Science Research Associates, 1971.
- Bohl, Marilyn, "Tools for Structured Design," Chicago, Science Research Associates, 1978.
- Bohl, Marilyn, "Essentials of Information Processing," Chicago, Science Research Associates, 1986.

- Buzy, Beth Moorer, **"Using Computers,"** Chicago, Science Research Associates, 1985.
- Chapin, N., **"Flowcharts,"** Princeton, Auerbach Publishers, 1971.
- Charette, Robert N., and Stockenberg, John, **"A Unified Methodolgy for Developing Systems,"** New York, McGrew-Hill Book company, 1986.
- Dahl, Olew Johan, and Others, **"Structured Programming,"** New York, Academic Press, 1972.
- Eliason, Alan L., **"Business Information Processing,"** Chicago, Science Research Associates, 1979.
- Essick, Edward L., **"Principles of Business Data Processing,"** 3rd ed., Chicago, Science Research Associates, 1986.
- Farina, Mario V., **"Flowcharting,"** Englewood Cliffs, Prentice-Hall, 1970.
- Gear, C. William, **"Introduction to Computers, Structured Programming, and Applications,"** Chicago, Science Research Associates, 1978.
- Gear, C. William, **"Computer Applications and Algorithms,"** Chicago, Science Research Associates, 1986.
- Keringham, Brian W., and Plauger, P. J., **"Elements of Programming Style,"** New York, McGrew-Hill Book Company, 1978.
- LaBudde, Keith P., **"Structured Programming Concepts,"** New York, McGrew-Hill Book Company, 1987.
- Lebtz, B. P., Swanson E. B., and Tompkins, G. E., **"Characteristics of Application Software Maintenance,"** Communication of the ACM, Vol. 21, No. 6 (June 1987).
- Leeson, Marjorie M., **"Programming Logic,"** Chicago, Science Research Associates, 1983.
- Leeson, Marjorie M., **"Computer Operations,"** 3rd., Chicago, Science Research Associates, 1987.
- Leeson, Marjorie M., **"Computer Information : A Modular System,"** Chicago, Science Research Associates, 1985.

- Linger, Richard C., and Others, **"Structured Programming : Theory, and Practice,"** Reading Mass., Addison-Wesley, 1979.
- Mader, Chris, **"Information Systems,"** Chicago, Science Research Associates, 1979.
- Martin, James, and McClure, Carma L., **"Structured Techniques for Computing,"** Englewood Cliffs, Prentice-Hall, 1985.
- Maurer, Ward D., **"Programming : An Introduction to Computer Techniques,"** San Francisco, Holden-Day, 1972.
- Maynard, Jeff, **"Modular Programming,"** Princeton, Auerbach Publisher, 1972.
- McGrower, Clement L., and Kelly, John R., **"Top-Down Structured Programming Techniques,"** New York, Petrocelli/Charter, 1975.
- O'Leary, T. J., and Williams, B. K., **"Computers and Information Processing,"** Menlo Park CA, Benjamin/Cummings, 1985.
- Radlow, James, **"Computers and the Information Society,"** New York, McGraw-Hill Book Company, 1986.
- Ralston, Anthony, **"Introduction to Programming and Computer Science,"** New York, McGraw-Hill Book Company, 1971.
- Rothman, Stanley, and Mosmann, Charles, **"Computer Uses and Issues,"** Chicago, Science Research Associates, 1985.
- Sanders, Donald H., **"Computer Concepts and Applications,"** New York, McGraw-Hill Book Company, 1987.
- Sanders, Donald H., **"Computers Today,"** New York, McGraw-Hill Book Company, 1985.
- Shelly, Gary B., and Cashman, Thomas J., **"Computer Fundamentals with Application software,"** Boston, Boyd & Frazer Publishing Company, 1986.
- Spencer, Donald D., **"Computers and Information Processing,"** Columbus, Merrill Publishing Company, 1985.  
 Spencer, Donald D., **"Computers : An Introduction ,"** Columbus, Merrill Publishing Company, 1986.

- Topping, Anne L., and Gibbons, Ian, **"Programming Logic : Structured Design,"** Chicago, Science Research Associates, 1985.
- Welsh, James, and McKeag, R. M., **"Structured System Programming,"** Englewood Cliffs, Prentice-Hall, 1980.
- Wetzel, Gregory F., and Bulgren, William G., **"The Algorithmic Process: An introduction to Problem Solving,"** Chicago, Science Research Associates, 1985.
- Edward Yourdon, **"Techniques of Program Structure and Design,"** Englewood Cliffs, Prentice-Hall, Inc., 1975.
- Yourdon, Edward, **"Managing the Structured Techniques,"** 3rd ed., New York, McGraw-Hill Book Company, 1988.

### ب — البرمجة بلغة بيسك :

- Alonso, J. R. F., **"Simple, BASIC Programs for Business Applications,"** Englewood Cliffs, Prentice-Hall, 1981.
- Benton, Stan, and Weeles, Len, **"Program It Right : Structured Methods in BASIC,"** Englewood Cliffs, Prentice-Hall, 1985.
- Bogart, Theodore F., **"Applied BASIC for Technology,"** Chicago, Science Research Associates, 1984.
- Boggs, Roy A., **"Applied BASIC for Microcomputer,"** Reston, VA, Reston Publishing Co., 1984.
- Bollot, Michel H., **"BASIC Concepts and Structured Problem Solving,"** St. Paul, West Publishing Co., 1984.
- Bosworth, Bruce, **"Business Programming Projects with BASIC,"** Chicago, Science Research Associates, 1984.
- Bradley, Julia C., **"MICROSOFT BASIC Using Modular Structure,"** Dubuque, Iowa, Wm. C. Brown Publishers, 1988.
- Chays Ruth K. and Miller, Joan M., **"More BASIC Programming for the Classroom and Home Teacher,"** New York, Teachers College Press, 1985.
- Compusoft, and Lien, David A. **"BASIC Programming for the IBM Personal Computer,"** Dubuque, Iowa Wm. C. Brown Publishers, 1984.

- Cox, Michael J., and Sullivan, Kathleen B., **"Structuring Programs in MICROSOFT BASIC,"** Boston, Boyd & Frazer Publishing Company, 1987.
- Dock, V. Thomas, **"BASIC Programming for Business,"** St. Paul, West Publishing Co., 1977.
- Ettlin, Walter A., and Solberg, Gregory, **"The MICROSOFT BASIC Book : Macintosh Edition,"** New York, McGraw-Hill Book Company, 1985.
- Finkel, LeRoy, and Brown, Jerald, **"Data File Programming in BASIC,"** New York, John Wiley and Sons, 1981.
- Goldstein, Larry J., **"IBM PC : An Introduction to the Operating System, BASIC Programming,"** 3rd ed., Bowie, MD, R. J. Brady Co., 1986.
- Gorsel, Jerome R., and Hills, William F., **"Debugging Techniques for IBM PC BASIC,"** New York, Brady Communication Co., 1986.
- Grame, Carl A., and O'Donnell, Daniel J, **"Learning BASIC,"** Chicago, Science Research Associates, 1984.
- Grout, Jarrell C., **"Programming with BASIC : A Structured Approach,"** Dubuque, Iowa Wm. C. Brown Publishers, 1985.
- Hearn, Donald, and Baker, M. Pauline, **"Computer Graphics for the IBM Personal Computer,"** Englewood Cliffs, Prentice-Hall, 1983.
- Hergert, Douglas, **"MICROSOFT Quick BASIC : Developing Structure Programs,"** Redmond, Wash., Microsoft Press, 1987.
- High-Speed Sorting Algorithm, "Communications of the ACM, July 1959, Vol. 2, PP. 30-32.
- Hirschfelder, R., and Others, **"Structure Vax BASIC,"** Menlo Park CA, Benjamin/Cummings, 1987.
- IBM Personal Computer Hardware Reference Library BASIC, P. 39.
- James, Mike, **"Artificial Intelligence in BASIC,"** Boston, Butlerworth, 1984.
- Keogh, James E., **"Programmer's Notebook : Modular Programming for Home Computers,"** New York, Simon and Schuster, 1984.



- Kittner, M., and Northcutt, B., "**BASIC : A Structured Approach**," Menlo Park CA, Benjamin/Cummings, 1987.
- Kittner, M., and Northcutt, B., "**Basic BASIC : A Structured Approach**," 2nd ed., Menlo Park CA, Benjamin/Cummings, 1987.
- Ledger, Henry, and Singer, Andrew, "**Elementary BASIC**," Chicago, Science Research Associates, 1982.
- Lesser, Murray L., "**Using the MICROSOFT Business BASIC Compiler on the IBM PC**," New York, McGraw-Hill Book Company, 1986.
- Lien, David A., "**BASIC Handbook : Encyclopedia of the BASIC Computer Language**," 3rd ed., San Diego, Compusoft, 1986.
- Orilla, Lawrence S., "**Structured BASIC : An Integrated Approach**," New York, McGraw-Hill Book Company, 1985.
- Peckham, Herbert, "**Hands-on BASIC for the Apple II Plus Computer**," New York, McGraw-Hill Book Company, 1984.
- Peckham, Herbert, "**Hands-on BASIC for the DEC Professional**," New York, McGraw-Hill Book Company, 1985.
- Peckham, Herbert, and Others, "**Structured BASIC for the IBM PC : A Hands-on Approach**," New York, McGraw-Hill Book Company, 1985.
- Peckham, Herbert, "**Hands-on BASIC for the Apple IBM Personal Computer**," New York, McGraw-Hill Book Company, 1983.
- Peckham, Herbert D., "**Hands-on BASIC for the Commodore 64**," New York, McGraw-Hill Book Company, 1984.
- Peckham, Herbert D., "**Intermediate BASIC for the TI Home Computers**," Englewood Cliffs, McGraw-Hill, Book Company, 1979.
- Peckham, Herbert D., "**Hands-on BASIC for the TRS-80 Color**," Englewood Cliffs, McGraw-Hill, Book Company, 1983.
- Peckham, Herbert D., "**BASIC : A hands-on Method**," 2nd ed., Englewood Cliffs, McGraw-Hill, Book Company, 1981.

- Peckham, Herbert D., **"MS DOS Structured BASIC,"** Englewood Cliffs, McGraw-Hill, Book Company, 1985.
- Quasney, James S, and Maniotes, John, **"Applesoft BASIC Fundamentals and Style,"** Boston, Boyd & Frazer Publishing Company, 1987.
- Quasney, James S, and Maniotes, John, **"BASIC Fundamentals and Style,"** Boston, Boyd & Frazer Publishing Company, 1987.
- Quasney, James S, and Maniotes, John, **"Complete BASIC for A Short Course,"** Boston, Boyd & Frazer Publishing Company, 1987.
- Quasney, James S, and Maniotes, John, **"Structured BASIC Fundamentals and Style for the IBM PC and Compatibles,"** Boston, Boyd & Frazer Publishing Company, 1988.
- Ruder, Jesse H., and Millsap, Gary, **"BASIC fo the IBM PC,"** New York, John Wiley and Sons, 1987.
- Ruder, Jesse H., **"BASIC fo the HP 3000,"** New York, John Wiley and Sons, 1986.
- Sawatzky, Jasper, and Chen, Shu-jen, **"Programming in BASIC PLUS,"** 2nd ed., New York, John Wiley and Sons, 1985.
- Scheinder, David I., **"Programming and Its Applications : Using IBM PC BASIC,"** San Francisco, Dellen Publishing Co., 1986.
- Shelly, Gary B., and Cashman, Thomas J., **"BASIC for the IBM PC,"** Boston, Boyd & Frazer Publishing Company, 1988.
- Simpson, Alan, **"Data File Programming on Your IBM PC,"** Berkeley, SYBEX, 1984.
- Sondak, Norman, and Hatch, Richard, **"Using BASIC on the CYBER,"** Chicago, Science Research Associates, 1982.
- Sondak, Norman, and Hatch, Richard A., **"Using BASIC on the IBM Personal Computer,"** Chicago, Science Research Associates, 1985.
- Spear, Bob, **"BASIC: Programming Fundamentals and Applications,"** Columbus, Merrill Publishing Company, 1987.
- Sternberg, Charles D., **"IBM Programs for Business,"** Rochelle NJ, Hayden Book Co., 1983.

- Waite, Mitchell, and Pardee, Michael, "**BASIC Primer**," Indianapolis, H. W. Sams, 1978.
- Wells, Timothy D., "**A Structured Approach to Building Programs : BASIC**," Englewood Cliffs, Prentice-Hall, 1985.
- Worland, Peter B., "**Introduction to BASIC Programming : A Structured Approach**," Boston, Houghton Mifflin, 1979.
- Zage, Wayne M., "**Programming with MICROSOFT BASIC**," Englewood Cliffs, McGraw-Hill Book Company, 1985.

«حقوق الطبع والنشر محفوظة لمعهد الإدارة العامة ولا يجوز إقتباس جزء من هذا الكتاب أو إعادة طبعه بأية صورة دون موافقة كتابية من إدارة البحوث إلا في حالات الاقتباس القصيرة بغرض النقد والتحليل مع وجوب ذكر المصدر»



## ● المؤلفان في سطور ●

### ■ الدكتور غازي إسحق الخطيب

- من مواليد عمان — الأردن .
- حصل على درجة الدكتوراه في الحاسب الآلي من جامعة ميسيسيبي ، ولاية ميسيسيبي عام ١٩٨٤م .
- يعمل حاليا أستاذا مشاركا — جامعة هاوارد — واشنطن .
- من خبراته العملية :
- أستاذ مساعد — معهد الإدارة العامة .
- أستاذ مساعد — جامعة ميسيسيبي — ولاية ميسيسيبي — أمريكا .
- من أهم أعماله العلمية المنشورة :
- مقارنة بين قواعد البيانات من ناحية الأمن والسرية — المؤتمر والمعرض الوطني التاسع للحاسب الآلي — الرياض — ١٤٠٧هـ . بالاشتراك مع عزمي العيسى .
- دور الحاسب الآلي في التنمية الإدارية — العدد (٥٠) — مجلة الإدارة العامة — معهد الإدارة العامة — الرياض بالاشتراك مع د . إبراهيم عبدالسلام ، وعبدالعزیز القويز .
- تقييم أداء الحاسبات الآلية في المراكز الحكومية في المملكة العربية السعودية — ندوة — معهد الإدارة العامة بالاشتراك مع د . إبراهيم عبدالسلام ، وكامل المبارك .

### ■ الأستاذ عزمي حسن العيسى

- من مواليد ارتاح — فلسطين .
- حصل على درجة الماجستير في الحاسب الآلي (تحليل وتصميم النظم) من جامعة بتسبرج — ولاية بتسلفانيا عام ١٩٨٤م .
- يعمل حاليا محاضرا بمعهد الإدارة العامة .
- من خبراته العملية :
- محلل نظم ، شركة افانثار الاستشارية لتنظيم المعلومات بتسبرج .
- محاضر ، كلية بوينت بارك — بتسبرج .
- مبرمج ، المجلس الوطني للتخطيط — عمان — الأردن .
- مشغل ، البنك التجاري الكويتي — الكويت .
- من أهم أعماله العلمية المنشورة :
- مقارنة بين قواعد البيانات من ناحية الأمن والسرية — المؤتمر والمعرض الوطني التاسع للحاسب الآلي — الرياض — ١٤٠٧هـ . بالاشتراك مع د . غازي الخطيب .
- نحو تصميم شبكة اتصالات بأقل التكاليف في المملكة العربية السعودية — المؤتمر والمعرض الوطني التاسع للحاسب الآلي — الرياض ١٤٠٧هـ . بالاشتراك مع د . نزيه الدريني وعبدالله باتو .



